

WILEY

Intl. Trans. in Op. Res. 31 (2024) 274–295
DOI: 10.1111/itor.13135INTERNATIONAL
TRANSACTIONS
IN OPERATIONAL
RESEARCH

A reinforcement learning approach for transaction scheduling in a shuttle-based storage and retrieval system

Banu Y. Ekren^{a,b,*} and Bartu Arslan^c^a*Department of Industrial Engineering, Yasar University, No: 37-39, Bornova, Izmir, Turkey*^b*School of Management, Cranfield University, UK*^c*Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, Eindhoven, the Netherlands**E-mail: banu.ekren@yasar.edu.tr [Ekren]; b.arslan@tue.nl [Arslan]*

Received 27 April 2021; received in revised form 11 November 2021; accepted 24 February 2022

Abstract

With recent Industry 4.0 developments, companies tend to automate their industries. Warehousing companies also take part in this trend. A shuttle-based storage and retrieval system (SBS/RS) is an automated storage and retrieval system technology experiencing recent drastic market growth. This technology is mostly utilized in large distribution centers processing mini-loads. With the recent increase in e-commerce practices, fast delivery requirements with low volume orders have increased. SBS/RS provides ultrahigh-speed load handling due to having an excess amount of shuttles in the system. However, not only the physical design of an automated warehousing technology but also the design of operational system policies would help with fast handling targets. In this work, in an effort to increase the performance of an SBS/RS, we apply a machine learning (ML) (i.e., *Q*-learning) approach on a newly proposed tier-to-tier SBS/RS design, redesigned from a traditional tier-captive SBS/RS. The novelty of this paper is twofold: First, we propose a novel SBS/RS design where shuttles can travel between tiers in the system; second, due to the complexity of operation of shuttles in that newly proposed design, we implement an ML-based algorithm for transaction selection in that system. The ML-based solution is compared with traditional scheduling approaches: first-in-first-out and shortest process time (i.e., travel) scheduling rules. The results indicate that in most cases, the *Q*-learning approach performs better than the two static scheduling approaches.

Keywords: SBS/RS; reinforcement learning; *Q*-learning; simulation; warehousing; automated storage

1. Introduction

With the recent Industry 4.0 developments established on the collaboration of automated systems within facilities, warehouse managers have become very eager to invest in automation technologies

*Corresponding author.

© 2022 The Authors.

International Transactions in Operational Research published by John Wiley & Sons Ltd on behalf of International Federation of Operational Research Societies

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

for their enterprises. For instance, they tend to deploy goods with radio frequency identification (RFID) tags, sensors, and other novel technologies, helping the implementation of robotic technologies to transform material handling technology through the transparency and traceability of products and smart environments.

According to ReportLinker (2020), the global automated material handling market (GAMHM) was valued at 50,544.6 million USD in 2019 and is expected to reach 101,487.3 million USD by 2025. In that GAMHM, the automated storage and retrieval system (AS/RS) market size was valued at \$7351 million in 2019, and it is projected to reach \$12,928 million by 2027 (Allied Market Research, 2020). AS/RSs are utilized for inventory management systems in warehouses, distribution centers, and manufacturing facilities. They work on computer-controlled systems for autonomous storage and retrieval of loads from one location to another. These systems are mostly utilized for large warehouses for quick, accurate, reliable, and low-cost solutions.

AS/RS technology providers aim to market designs providing maximized storage capacity, increased performance, reduced energy costs, and increased inventory accuracy and customer service. For instance, one of the well-known companies, Dematic Group, markets a mini-load AS/RS technology providing ultrahigh-speed load handling referred to as shuttle-based storage and retrieval system (SBS/RS). SBS/RS is mostly utilized in large distribution centers and in raw material storage places that are specially designed for ultrahigh-speed load handling (Dematic, n.d.). Due to having a dedicated shuttle in each tier of the shelves, this design is also referred to as the tier-captive SBS/RS in the literature.

To describe, SBS/RS is an AS/RS that comprise shelves, shuttles, and lifts. Shuttles and lifts are fully automated devices. In an initial design implementation of SBS/RS, shuttles are designed to be tier-captive, providing horizontal movement for loads within a tier. A lifting mechanism is installed at each cross-aisle, providing vertical movement for loads. Lifts carry storage transactions for their destination tiers from the input/output (I/O) locations located at the first (ground) level of each aisle. Lifts also carry the retrieval transactions from their current tiers to the I/O locations in their aisle addresses. Shuttles pick up/drop off the transactions at the buffer locations located at both sides of each tier connected with the lifting mechanism.

Figure 1 shows side and top views for a tier-captive SBS/RS design. For instance, there are eight tiers in that design. Lift 1 represents the lifting mechanism located at the cross-aisle that can carry two loads independently. As mentioned, we refer to this design as the tier-captive SBS/RS throughout this paper. Note that there is a dedicated shuttle in each tier in this design.

A typical handicap of a tier-captive SBS/RS design is that, since there is a dedicated shuttle in each tier of shelves and a single lifting mechanism installed at each cross-aisle, lifts mostly become bottlenecks in the system. Namely, the average utilization of shuttles is very low, compared to those lifts. In an effort to overcome this handicap, we propose an alternative SBS/RS design in which the total number of shuttles is decreased in the system so that shuttles can travel between tiers within a dedicated aisle (see Fig. 2). With the decreased number of shuttles in the system, we aim to balance the utilization of shuttles and lifts as well as have a decreased investment cost in the system. Here, shuttles can travel between tiers by using a separate lifting mechanism installed at the other cross-aisle side. The new lifting mechanism is solely dedicated to the travel of shuttles between tiers. The applicability of transportation of autonomous vehicles between tiers by a lifting mechanism has already well-studied by Ekren et al. (2010), Ekren and Heragu (2010a), Ekren and Heragu (2011), Ekren et al. (2013), and Ekren et al. (2014). The advantage of this new design would

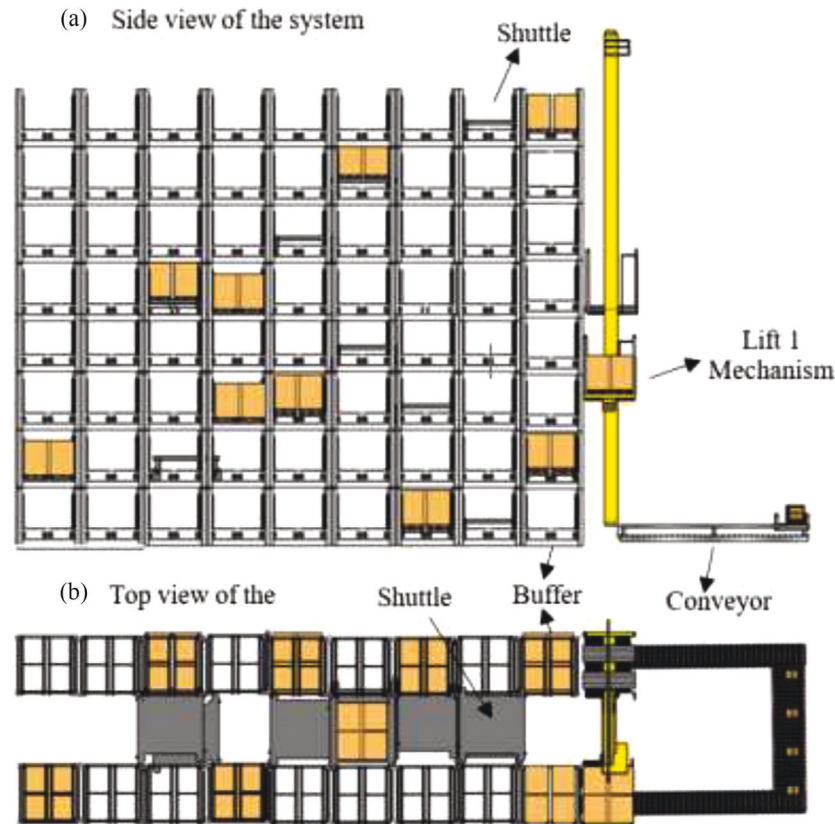


Fig. 1. A tier-captive shuttle-based storage and retrieval system (SBS/RS) design.

be the decreased number of shuttles in the system also resulting in decreased initial investment cost as well as increased utilization of shuttles. We refer to this novel design as the tier-to-tier SBS/RS throughout this paper and show this proposed system in Fig. 2.

Remember that while the tier-captive SBS/RS has a single lifting mechanism dedicated to carry loads, in the proposed tier-to-tier SBS/RS, there are two lifting mechanisms, one of which is dedicated to the vertical travel of loads (i.e., totes). This lifting mechanism is referred to as Lift 1 in Fig. 2. The other lifting mechanism dedicated to the travel of shuttles between tiers is referred to as Lift 2 in Fig. 2. Loads are picked up/dropped off by the shuttles at the buffer locations for storage/retrieval of transactions. If the process is a retrieval process, then the load is dropped off at the I/O point (i.e., the ground level) by Lift 1.

Due to the flexible travel of shuttles within an aisle, one of the significant management issues in a tier-to-tier SBS/RS design is selecting the most proper transaction waiting in the shuttle queue. Namely, when a shuttle becomes idle, it should select the most proper transaction waiting in its queue so that the system's transaction process rate increases. In the studied tier-to-tier SBS/RS, because there are three different servers (i.e., Lift 1, Lift 2, and shuttles) and they are working interactively with each other as well as each server's process time might be affected by a transaction

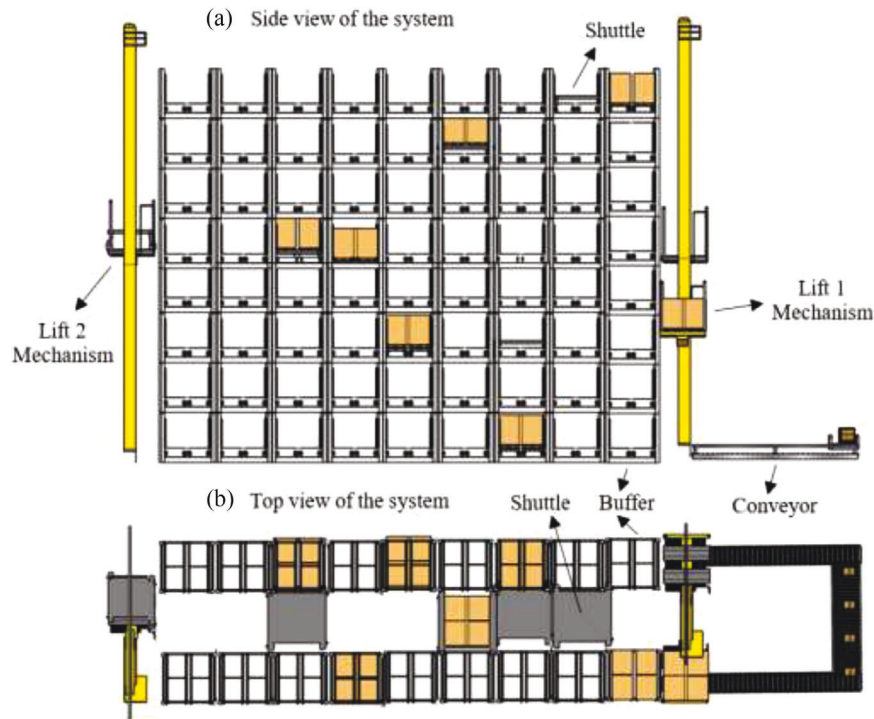


Fig. 2. A tier-to-tier SBS/RS.

type (i.e., storage or retrieval) processed and its address, it is significant to allocate the most proper transaction to the available shuttles resulting in decreased travel time for each server. Hence, in such an integrated and complex operating system, developing smart management algorithms resulting in the increased performance of the system becomes an important issue. For this purpose, we propose a recent machine learning (ML) algorithm, the Q -learning algorithm, for the smart transaction selection problem in this system. Instead of a static scheduling rule, we suggest a trial-and-error method (i.e., reinforcement learning [RL] algorithm) where intelligent shuttle agents are trained during their practical experiences in the system. The RL algorithm is modeled by simulating the system in ARENA 16.0, a commercial software. The results are compared with a first-in-first-out (FIFO) and shortest process time (SPT) scheduling rules. Details of the RL algorithm as well as the implementation procedure are presented in Section 3.

The main research questions in this paper can be summarized as follows:

- Q1–Is there an alternative design for SBS/RS in which shuttle and lift utilizations are balanced?
- Q2–Can an ML method be applied for smartly operating (i.e., selecting) transactions in the proposed novel SBS/RS design, resulting in better performance metrics than some well-known static rules?

By Q1, we suggest a novel SBS/RS design that is the tier-to-tier SBS/RS (see Fig. 2). By Q2, we propose an RL algorithm developed on Q -learning.

ML-based methods are gaining popularity among researchers and industry (Bengio et al., 2021; Emilio et al., 2021). Bertolini et al. (2021) review the papers that are based on ML and deep learning approaches applied to industry problems. They discover that production planning and control and defect analysis are the most studied subjects. Their study suggests that more ML-based methods will be common in the field of operations management.

In order to handle today's major challenges of complex manufacturing systems, ML techniques play a significant role, which is data-driven approaches, and are able to find highly complex and non-linear patterns in environmental data. They can be later applied for prediction, detection, classification, regression, or forecasting. By applying an ML-based modeling to an automated warehousing problem, this paper aims to be a pioneer in the field of warehouse management implementing artificial intelligence (AI) methods. Additionally, successful applications of RL methods on dynamic task scheduling are presented well in Shyalika et al. (2020).

The flow of the paper is as follows: Section 2 describes the literature review. Section 3 presents the implementation of the RL algorithm on the proposed SBS/RS design. Section 4 gives the results and comments on the implementation of the methodology. Section 5 summarizes the paper along with the findings.

2. Literature review

In this section, we present a literature review in two sub-sections. First, we summarize the current works on SBS/RS and its variants. Second, because we focus on ML application on transaction scheduling in the system, we summarize the ML-based works on task scheduling.

2.1. SBS/RS-related works for tier-captive design

Most studies in the literature focus on tier-captive SBS/RS. We discuss both tier-captive and tier-to-tier-related studies in this section.

Marchet et al. (2011) model tier-captive SBS/RS design by an open queuing network modeling approach. Their aim is to estimate the average waiting time as well as the average cycle time per transaction performance metrics correctly. Later, Marchet et al. (2013) show design trade-offs in the tier-captive SBS/RS by using a simulation modeling approach. Several designs are evaluated by considering different performance metrics, including the initial investment cost. The results suggest that decreasing the number of aisles provides decreased investment costs in the system.

Carlo and Vis (2012) study a novel variant of a tier-captive SBS/RS utilized by a company in the Netherlands. There are non-passing lifting mechanisms in the system. They develop heuristic-based scheduling algorithms developed on two functions for scheduling those lifts.

Lerher et al. (2015b) show the advantages of the tier-captive SBS/RS by comparing it with another warehousing technology. They focus on throughput rate performance metrics for different designs of the warehouse. They show that SBS/RS produces a higher throughput rate. Lerher (2015) studies double-deep storage compartments in a tier-captive SBS/RS. The advantage of this racking system is that the floor space utilization is decreased by the decreased number of aisles in the

system. Lerher et al. (2015a, 2016) develop analytical travel time models for a tier-captive SBS/RS by including velocity profiles of shuttles and lifts (e.g., acceleration, deceleration, and maximum velocities) as well as single and dual command scheduling rules of transactions in the systems. The proposed models are validated by their simulation models. That work is lacking in considering queuing performance metrics. Ekren et al. (2015) study class-based storage policy in a tier-captive SBS/RS by using simulation modeling. They show the efficiency of the class-based storage policy by comparing it with a random storage policy.

Ekren and Heragu (2011) present simulation-based design approaches for a different variant of an automated warehousing technology. Ekren (2017) shows the design trade-offs of a tier-captive SBS/RS under different warehouse designs. She presents several graphs illustrating several performance metrics under those designs allowing warehouse managers to decide the most proper one based on their requirements. Ekren et al. (2018) present analytical models predicting the mean and variance of the travel time of lifts and shuttles in a tier-captive SBS/RS. In addition, those models can also predict the mean amount of energy consumption and energy regeneration per transaction based on several design parameters: velocity profiles of vehicles/lifts, number of tiers, number of bays, and so forth.

Wang et al. (2015) propose a mathematical model for scheduling tasks in a tier-captive SBS/RS. They apply a non-dominated sorting genetic algorithm (GA) in order to find out a multi-objective optimal solution. Tappia et al. (2016) present a queuing network model predicting several performance metrics from a tier-captive SBS/RS. Zou et al. (2016) propose an approximate model developed on a fork-join queueing network approach to estimate several performance metrics from a tier-captive SBS/RS. They consider the sequential movement and parallel processing of vehicles and lifts in the model. Recently, Eder (2019) studied a continuous-time open queueing network (OQN) with limited capacity to estimate important performance metrics from a tier-captive SBS/RS design.

Kazemi et al. (2019) propose a hybrid solution combining the ant colony algorithm and adaptive large neighborhood search. Their system design includes a storage and retrieval machine that can move simultaneously in horizontal and vertical directions. The system utilizes shuttles to move within aisles. In a recent work, Liu et al. (2021) study a tier-captive SBS/RS, and they propose a mathematical model including a cross-aisle shuttle that travels the transactions between aisles. They consider throughput rates, travel times, and energy consumptions. In addition, the optimal velocity and acceleration values are studied. Wauters et al. (2016) study a mini-load AS/RS design including a dual shuttle crane. They introduce a mathematical model and different heuristic strategies for the optimization of the system.

A recent study on tier-captive SBS/RS has been completed by Ekren (2020a). In that work, she studies a statistical experimental design to identify significant design factors affecting the performance of tier-captive SBS/RS. The paper shows that increasing the number of aisles in the system affects the system performance, significantly. Ekren (2020b) proposes a multi-objective optimization solution for the design of a tier-captive SBS/RS. In that study, as multi-objectives, minimization of both average cycle time per transaction, and energy consumption per transaction is considered. Ekren and Akpunar (2021) present a tool that can compute several performance metrics from a pre-defined SBS/RS design. The tool is developed on an open queueing network modeling approach and shared via a website for free.

2.2. SBS/RS-related works for tier-to-tier design

A similar system is also referred to as the autonomous vehicle-based storage and retrieval system (AVS/RS) in the literature, where autonomous vehicles can travel between tiers and aisles in a more flexible travel pattern in the system. We provide some of them here. Besides, there are few studies on tier-to-tier SBS/RS. We present all of them here.

Ekren and Heragu (2012) compare the performance of two systems in automated warehousing: crane-based AS/RS and AVS/RS. The performance of the systems is compared based on their initial investment costs and several other operational metrics. Ekren et al. (2013, 2014) model AVS/RS by semi-OQN models efficiently. In the solution of the proposed queueing network models, they utilize their pre-extended algorithm (Ekren and Heragu, 2010a). Heragu et al. (2011) study analytical models for crane-based AS/RS and AVS/RS by OQN models. They utilize the manufacturing system performance analyzer tool for modeling purposes.

The first study on the tier-to-tier SBS/RS is studied by Ha and Chae (2018a). They define a free-balancing approach for collision prevention of shuttles in the system. Their system includes a single lifting mechanism carrying both loads and shuttles. By also studying a non-free balancing approach, they compare the two solution approaches on the system performance. Later, Ha and Chae (2018b) introduce a decision model to determine the optimal number of shuttles in the tier-to-tier SBS/RS by using several parameters, such as acceleration, velocity, physical designs, and so forth.

Zhao et al. (2019) publish a study proposing an integer programming model for the scheduling of transactions to minimize the idle time of lifts and waiting time of shuttles in a tier-to-tier SBS/RS by utilizing simulation-based optimization. Their work lacks in ignoring collision possibilities as well as in considering both storage and retrieval processes in the system.

A recent work by Küçükyaşar et al. (2020) study the comparison of tier-captive and tier-to-tier SBS/RSs under different design configurations. In the comparison, the initial investment cost and some other performance metrics are considered. The results suggest that there could be better designs in tier-to-tier SBS/RS than in tier-captive SBS/RS with decreased investment cost and increased performance metrics. Later, Kucukyasar et al. (2021) study a tier-to-tier SBS/RS design from an energy efficiency design. Kucukyasar and Ekren (2022) also study the scheduling rules of transactions in a tier-tier SBS/RS by comparing two selected rules: the FIFO and SPT rules. Their results show that SPT outperforms FIFO. Later, Kucukyasar and Ekren (2022a) also show that the number of tiers, shuttles, and aisles affect the performance of the tier-to-tier SBS/RS design, significantly.

Arslan and Ekren (2021) apply deep Q -learning for a tier-to-tier SBS/RS for smart transaction selection of shuttles. By that, they aim to have system performance with decreased average cycle time per transaction performance metric. A different work is completed by Eroglu and Ekren (2022), where shuttles can travel flexibly between aisles within a tier. In that work, the authors focus on collision and deadlock prevention algorithms for safely travel of shuttles.

Different from the current works, this paper studies a tier-to-tier SBS/RS by implementing a novel RL-based scheduling procedure for smart transaction selection of shuttles from their queue. The performance of the system is observed by the average cycle time per transaction and the average waiting time of a transaction in the queue. Note that to the best of our knowledge, this is the first time that we have applied an ML-based approach for an AS/RS. In the following subsection, we summarize how RL-based scheduling algorithms are implemented on related cases.

2.3. RL-based automated guided vehicle (AGV)-related works

Scheduling and priority assignment problems are hot topics for multi-robotic systems in warehouses. To the best of our knowledge, there is no work studying intelligent task selection problems (e.g., by applying an ML-based algorithm) for autonomous storage and retrieval systems. However, there are few works on the application of ML algorithms for the dispatching of jobs in AGVs systems. We summarize the most related works here.

Jennings (1996) introduce a task negotiation approach for autonomous robots in warehousing systems. However, when the environment becomes stochastic, the approach cannot be applied. A combinatorial auction algorithm is proposed by Sandholm (2002). The algorithm divides tasks into different clusters based on correlations between them, and then, robots bid on the unrestricted combinations of tasks. This makes the solution more efficient than the traditional mechanisms.

A hybrid solution approach for task scheduling in an automated warehouse environment with multi-robot system is presented by Dou et al. (2015). They propose a GA-based task scheduling method combined with RL.

Xue et al. (2018) apply an RL algorithm for the flow-shop scheduling problem for a multi-AGV environment. The objective function considers the minimization of the average job delay and total makespan in the system. They model the problem as a Markov process by defining state features, action space, and reward function based on RL. Simulation results show that the RL agent can learn the optimal or near-optimal solution from its past experience, and the results provide better performance outputs than the multi-agent scheduling method.

Malus et al. (2020) present how the multi-agent RL method can be applied to the problem of order dispatching of autonomous mobile robot fleets. In the proposed problem, agents can be trained to bid for the orders to process given order specifications. The results show that the training policy outperforms the closest-first rule.

Watanabe et al. (2001) propose two methods providing intelligence for AGVs. The first is the AGV navigation problem, and the second is the collision avoidance problem. In the first problem, the navigation routes are learned by the AGVs with the help of Q -learning. The second problem is defined as the mutual understanding of behaviors among AGVs by using Q -learning. In the experimental simulations, it is verified that the two proposed methods can manage the target goals.

Takahashi and Tomah (2020) propose a deep RL method to control multiple AGVs optimally. Simulation experiments are performed to train the model. Later, the learned network is used to optimize another model. The results show that the proposed method learns optimally or near-optimally from past experience and provides superior performance in new environments.

Different from the existing studies, in this paper, we apply an RL algorithm developed on Q -learning for the shuttle's transaction selection problem. Here, we treat the shuttles as intelligent agents so that they can be trained from their past experiences. The modeling approach is detailed in the following section.

3. Simulation-based RL modeling

In this section, we present the developed RL algorithms for smart transaction selection of shuttle agents. First, we present the main working rule of the RL method. Second, since we model

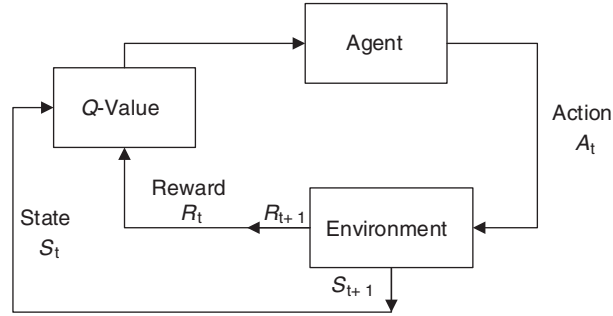


Fig. 3. Working rule of the reinforcement learning (RL) algorithm.

the system by simulation, we present the simulation modeling details along with the considered assumptions. Third, we present the RL implementation on the problem and the obtained results.

3.1. RL method

RL is an ML modeling approach that addresses how an autonomous agent is trained to select the most appropriate action by interacting with its environment to achieve its goal. In this framework, an agent sensing and observing its environment is created so that the current state changes are tracked in real time. By using the perceived state information, that agent aims to select an action maximizing its reward. When an action is performed by the agent, the current state may change. That state is rewarded or penalized based on the selected action. These interactions between the agent and its environment continue until the agent is trained for a decision-making strategy maximizing the total reward. To address an RL problem, Sutton and Barto (2015) define four key elements: a policy, reward function, value function, and the model of the environment. Here, the intelligent RL agent aims to maximize some portion of the defined cumulative reward. Figure 3 shows the working rule of the RL algorithm. According to that, there is a continuous interaction between the environment and the agent. Based on the changed environment, the agent takes an action at time t .

Some important terms that are used in the RL algorithm are summarized below:

- *Agent*: It is an entity performing actions in an environment.
- *Environment*: A set of states that an agent has to face.
- *Reward (R)*: An immediate return (i.e., performance metric) given to an agent when it performs a specific action or task.
- *State (s)*: Current condition returned by the environment.
- *Policy (π)*: A strategy applied by the agent to decide the next action based on the current state.
- *Value (V)*: Expected long-term return with discount compared to the short-term reward.
- *Value Function*: It specifies the value of a state that is the total amount of reward

In this study, we evaluate the value (V) of the system by the average cycle time of a transaction. Here, cycle time is the time between when a transaction request is created in the system until it is

disposed. We define the Q value (or action value) by (1), which is calculated by taking an additional parameter as a current action. In Q -learning, the aim is to develop a state-action matrix by providing a value function regarding the system environment. The main goal of this is to train the agent so that it can select the best policy (action) under a current state. Remember that the state includes all the information about the environment that is needed to take an action and that an action is what to do in a given state.

$$Q_{new}(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha [R(s, a) + \gamma \max_{a'} Q(s', a')]. \quad (1)$$

In (1), $Q_{new}(s, a)$ is the newly derived Q value, α is the learning rate, γ is the discount rate, and $R(s, a)$ is the reward when action a is taken in state s (Gosavi, 2015). $\max Q(s', a')$ is the maximum expected future reward among all possible actions at the next state s' , when action a is taken at state s .

Note that RL is a “model-free” algorithm not using the transition probability distribution and the reward function associated with the Markov decision process (Gosavi, 2015). Instead, RL finds out those distributions and functions by trial-and-error. Namely, the transition probability distribution and the reward function together are called the “model” of the environment, and they are estimated by the RL algorithm in simulation.

In the working principle of Q -learning, after a sufficient amount of time, the Q -matrix becomes stable. This means that the intelligent agent is trained so that it estimates the state-action relationship based on the reward output. In other words, the system agent can estimate how big a reward can be earned when and afterward of a state.

In the tier-to-tier SBS/RS, the environment (e.g., state (s of the system) is determined by (i, j), where i and j show the current tier of Lift 1 and the current tier of the available shuttle, respectively. The actions are defined based on the attributes of the waiting transactions. Waiting transactions have several attributes, two of which are significant and might affect the processing time of the servers. These are: the tier addresses of transactions and the transaction type (i.e., storage or retrieval). Thus, the actions (a) in the Q -matrix are defined to be (a_1, a_2), where a_1 and a_2 denote the tier address and the transaction type of the waiting transactions. The state-space of transaction types is defined as $\{1, 2\}$ where 1 represents storage and 2 represents retrieval transaction types. Once again, by the described actions, the aim is to select a waiting transaction providing the best reward (i.e., decreased cycle time). For modeling purposes, first, we simulate the system, and then we integrate the RL algorithm into the simulation model. In the following sub-section, the details of the simulation model and how the RL algorithm is integrated into this model are presented.

3.2. Simulation modeling of the system

Due to the operational complexity of the proposed SBS/RS design, where shuttles can travel between tiers by using a separate lifting mechanism and there is a collision possibility of shuttles, as well as there are mainly three servers (i.e., Lift 1, Lift 2, and shuttles) working interactively with each other, a simulation modeling approach is utilized. Before implementing the RL algorithm, first, the considered system is simulated by the ARENA 16.0 commercial software. The notations that we use in this paper are summarized in Table 1.

Table 1
The notations used for the model

Notation	Description	Unit
$VS_{\max}$	Maximum velocity that shuttles can reach	m/s
as_v	Acceleration value for the shuttles	m/s ²
ds_v	Deceleration value for the shuttles	m/s ²
$VL_{\max}$	Maximum velocity that lifts can reach	m/s
al_v	Acceleration value for the lifts	m/s ²
dl_v	Deceleration value for the lifts	m/s ²
L	Distance between two adjacent bays	m
H	Distance between two adjacent tiers	m
US_{avg}	Average utilization of a shuttle	%
UL_{avg}	Average utilization of Lift 1	%
ULS_{avg}	Average utilization of Lift 2	%
C_{avg}	Average cycle time of a transaction	s
W_{avg}	Average waiting time of a transaction	s
W	Average number of transactions waiting in shuttle queue	
T	Number of tiers in a single aisle	
B	Number of bays at a single side of a tier in an aisle	
N_S	Number of total shuttles in the system	

For a storage transaction, those processes take place:

1. Lift 1 travels to the I/O point (i.e., first tier) to pick up the arriving tote. At the same time, the shuttle travels to the Lift 2 location to travel to the storage tier address if it is not at that tier.
2. Lift 1 travels to the storage address tier and drops off the tote at the buffer location.
3. The shuttle travels to the buffer location to pick up the tote at the storage tier.
4. The shuttle travels to the storage bay with the tote and drops off it at the storage bay.

For a retrieval transaction, those processes take place:

1. The shuttle travels to the Lift 2 location, to travel to the retrieval tier address, if it is not at that tier.
2. Lift 2 travels to the shuttle's tier, and both travel to the retrieval tier.
3. The shuttle travels to the retrieval bay address to pick up the tote at the retrieval address.
4. The shuttle travels to the buffer location to drop off the tote. Lift 1 travels to the retrieval address tier to pick up the tote.
5. Tote travels to the I/O point in Lift 1.

The simulation model's flow chart including the RL algorithm is given in Fig. 4.

First, we verify and validate the simulation models by debugging the codes. Then, the RL algorithm is integrated by coding in the visual basic for applications (VBA) interface in this software. The considered operating rules along with the tier-to-tier SBS/RS assumptions are provided below:

- Transactions arrive at the I/O locations in the SBS/RS warehouse and enter a single shuttle queue defined for all shuttles immediately.

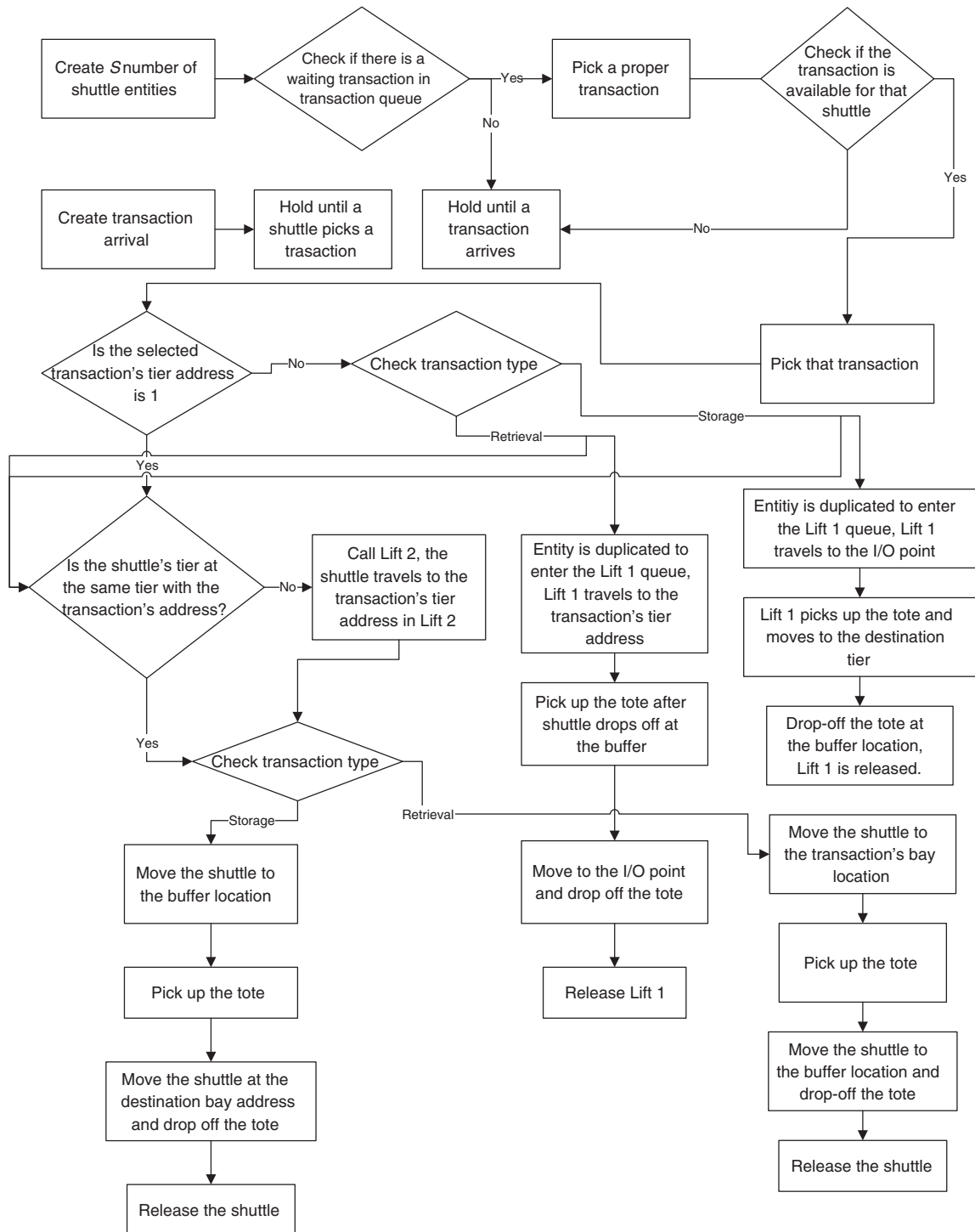


Fig. 4. Flow chart of the RL algorithm and the simulation model.

- Shuttles are defined as agents. When a busy shuttle becomes available, the environment (state) changes. The available shuttle agent selects the best transaction from waiting transactions based on the RL rule defined by the Q -matrix (see Section 3.2).
- In an effort to avoid collision of shuttles, the available shuttle agent first eliminates the transactions at whose tier addresses, there are already shuttles or there is a shuttle heading to that tier.
- Once the available shuttle selects a waiting transaction from the queue, the entities are duplicated and are immediately sent to the Lift 1 and Lift 2 queues based on the requirements. Hence, simultaneous travel of lifts and shuttles is allowed in the system. To avoid losing the future (i.e., next state's) tier address information, the queue order of lifts is considered to be a FIFO rule.
- If the transaction is a storage process, then Lift 1 transfers the load to the storage address. If the transaction is a retrieval process then, Lift 1 carries the load at the ground level. If the transaction address is at the first (ground) level, then Lift 1 is not utilized.
- Lift 2 is utilized when the shuttle requires changing its tier level. If Lift 2 is seized during the shuttle's travel, the two can travel simultaneously.
- Lift 1 has two loads capacities that can travel independently. Lift 2 has a single shuttle capacity.
- The simulation stops when there is no change in the average cycle time per transaction performance metric in a pre-specified time amount.
- As a dwell point policy, shuttles and lifts are assumed to stay at their last process points.

For the data used as model inputs, we use parameters from Lerher et al. (2015a, 2015b), Ekren et al. (2018), and (Ekren, 2020a).

- $L = 0.5$ m.
- $H = 0.35$ m
- $VS_{\max} = VL_{\max} = 2$ m/s
- $as_v = ds_v = al_v = dl_v = 2$ m/s²
- $B = 25$
- Arrivals of transactions at the warehouse follow a Poisson process.

By assuming a random storage policy in the system and that each aisle is identical in terms of the number of tiers and bays and arriving loads, we simulate a single aisle. Due to the Poisson arrival assumption, to calculate the mean arrival rate to a single aisle, we divide the mean arrival rate by the total number of aisles in the system. We detail the RL algorithm integrated into the simulation model in Section 3.2.

3.3. RL implementation

In the problem, we treat shuttles as agents. Before a shuttle picks a transaction from its queue, the current state information is observed from the environment.

We define the state as $S(k) = (i, j)$:

$S(k) = (\text{Current tier of Lift 1}, \text{Current tier of the agent shuttle } k)$

Here, k represents the shuttle ID selecting a transaction from the queue, i represents the current tier of Lift 1, and j represents the current tier of shuttle k . From the Q -table, first, the agent checks the current positions of the two Lift 1 lifting tables and then selects the lifting table with the higher Q -value. In this case, the state information takes integer values between 1 and T , in which T represents the total number of tiers in the system.

The actions are defined as the attributes of the waiting transactions.

$A(k) = (\text{Tier address of the transaction, transaction type})$

For agent shuttle k , the tier addresses of transactions are integers between 1 and T , and the transaction type can be either 0 (for storage) or 1 (for retrieval).

In the case of two shuttles available at the same time, priority is given to the first released shuttle to select an action (i.e., transaction). However, since the experiments are designed in such a way that average shuttle utilizations are very high, this case rarely occurs. Since the state-space consists of tiers of the lifts and shuttle and the action space consists of tier addresses and transaction types, the Q -matrix size is equal to the multiplication of states and actions, $T^2 \times 2T$. An example for $T = 5$ is given below.

$$Q = \begin{matrix} & \text{States} & \text{Actions} \\ \begin{matrix} (1, 1) \\ (1, 2) \\ (1, 3) \\ \vdots \\ (5, 4) \\ (5, 5) \end{matrix} & \begin{bmatrix} (1, 1) & (1, 2) & (2, 1) & (2, 2) & \dots & (5, 2) \end{bmatrix} \end{matrix} \quad (2)$$

Here, the reward function is defined by (3):

$$R(s, a) = \frac{1}{\text{flow time}(a)}. \quad (3)$$

Since the queue length and content information of transactions are not included in the state definitions, the reward is considered to be flow time, which does not include the waiting time value in it and does not correlate with the states. The steps of the considered RL algorithm are explained below.

Step 1. Initialize the Q -values. In other words, set for all (s, a) where $s \in S$ and $a \in A$: $Q(s, a) = c$.

Here, the initial values of c are calculated by solely considering the travel time of shuttles and lifts for each pair of state and action by ignoring any waiting times. Set $k = 0$, the number of days the program runs. We run the algorithm for $k_{\max}$ iterations, where $k_{\max}$ is chosen to be 40 days. Start the system simulation at an arbitrary state.

Step 2. Observe current state $S(i, j)$.

Step 3. With probability $1-\beta$, select random action a , otherwise select $a = \arg\max Q(s, a)$

Step 4. Execute action a . Let $R(i, j, a)$ be the immediate reward earned by state (i, j) under the influence of action a . Let the next state be $S(i', j')$

Step 5. Update $Q(s, a)$ by using (1)

Step 6. If $k < k_{\max}$, set be $(i, j) \leftarrow (i', j')$, go to Step 2, otherwise end the simulation.

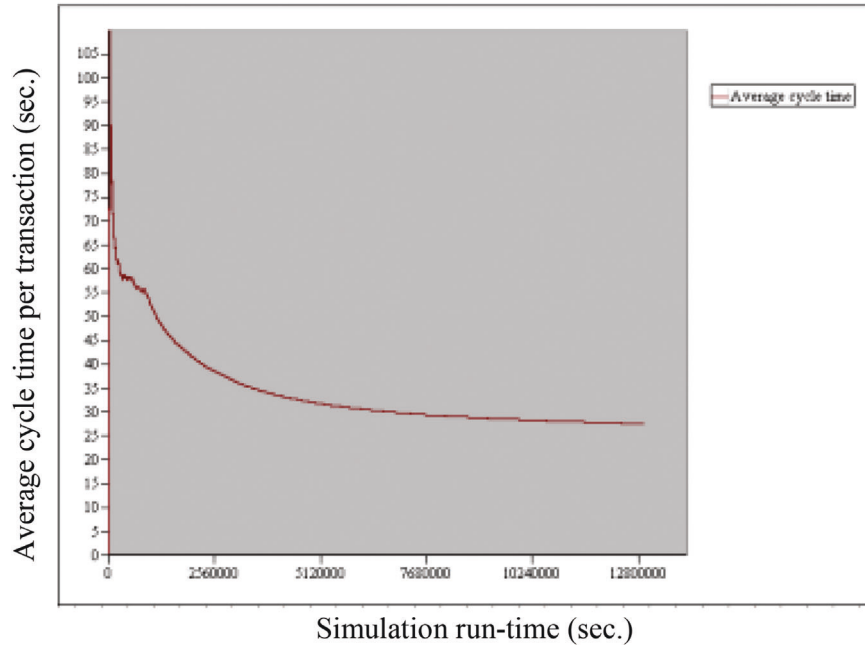


Fig. 5. Average cycle time per transaction change during training.

In the solution procedure, the β is initialized as 0.2, $\alpha = 0.1$, $\gamma = 0.2$. In the learning period, the β and α values are updated dynamically in time, based on the learning condition. Namely, the exploration probability (i.e., β value) and the learning rate is decreased in time because the agent learns the state-action relation based on the reward produced. Hence, the β value is increased and α is decreased by 0.2 and 0.02, respectively, every 10 days, until β becomes 1 and α becomes 0. Q -values are initialized by travel times of servers by considering that they are at middle locations through their travel paths. This would help decrease of training times of shuttle agents. Since the values are initialized and the system is stochastic, α value is considered to be low. In addition, since the next available actions are unknown, we keep the γ value at a low rate of 0.2.

The RL algorithm is coded by using the VBA interface integrated into the ARENA 16.0 software. After verifying and validating the model, we complete several experiments to test the behavior of the system developed on the ML operating rule.

Figure 5 shows the average cycle time per transaction versus time graph during the learning process. The time point where the cycle time is stable is the point at which the RL agent completes the learning process. Note that in Fig. 5, initially, the average cycle time per transaction is high and tends to decrease. This is because the RL agent is trained in time. The statistics in the simulation are collected after that time as shown in Fig. 6.

Since the initial Q -values are assigned based on each server's travel time from a middle point from its travel path, computational time tends to decrease, compared to assigning all the Q -values to 0. Even though we initialize the values based on those travel lengths, since the compared algorithms are static (i.e., FIFO and SPT), RL runs longer to converge than those compared models.

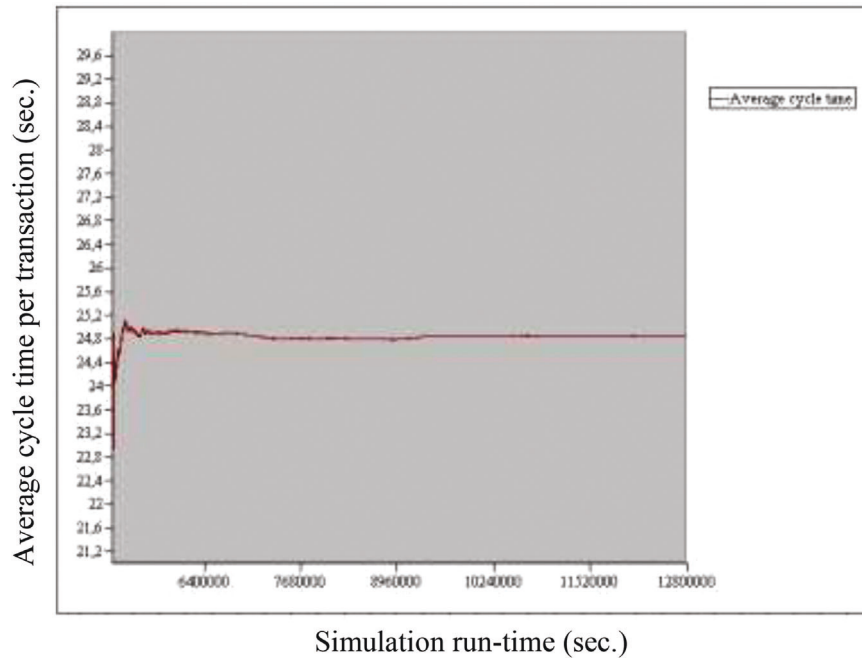


Fig. 6. Average cycle time per transaction after training.

In Fig. 5, the time point at which the RL agent completes its learning process is assumed to be approximately 3,000,000 seconds. Hence, statistics of interest are observed after that point for comparison purposes.

4. Experiments, the results, and comments

Recall that the immediate reward is computed based on the flow time of the processed transaction shown by (3). However, we also track the cycle of a transaction representing the time between when a transaction is created until its process is completed (i.e., until the entity is disposed). Note that during the process of a transaction, Lift 1, Lift 2, or both might be requested. Hence, the waiting time in those servers' queues is also included in the cycle time consideration. In the environment definition, since we do not include any information about the queue lengths of those Lift 1 and Lift 2 servers, as well as bay addresses, the same state-action pair may provide different cycle time values. In an effort to obtain a stable Q -matrix and ignore such unknown variations, we obtain the average of the previously earned rewards at the same pair action. Since the Q -matrix size increases exponentially by the increase of T in the system, we conduct experiments for the tier-to-tier SBS/RS having up to 15 number of tiers in each aisle. The conducted experiments are summarized in Table 2. According to that, we change T scenarios for 15, 12, and 10. For each T scenario, a specific number of shuttles per aisle scenario is also considered. For instance, for the 15-tier scenario, it is assumed that there are five shuttles per aisle. For instance, we try that combination for all the scheduling rule scenarios, FIFO, SPT, and RL.

Table 2
Conducted experiments

T – N_s	Scheduling rule	Arrival scenario
15–5	First-in-first-out (FIFO)	Single type
12–4	Shortest process time (SPT)	Batch
10–3	Reinforcement learning (RL)	

Table 3
Experimental results for single transaction arrival

Rule	T	N_s	UL_{avg}	ULS_{avg}	US_{avg}	C_{avg}	W_{avg}	Arrival rate (transaction/hour)
FIFO	15	5	87%	93%	90%	55.49 ± 0.46	36.98 ± 0.45	947
RL	15	5	83%	82%	81%	24.8	7.93	947
SPT	15	5	83%	81%	81%	24.28 ± 0.21	7.48 ± 0.09	947
FIFO	15	3	67%	79%	91%	61.59 ± 1.07	44.2 ± 1.06	600
RL	15	3	64%	70%	85%	28.21	11.86	600
SPT	15	3	64%	69%	85%	27.74 ± 0.06	11.43 ± 0.18	600
FIFO	12	4	78%	88%	93%	101.85 ± 3.5	84.8 ± 3.5	837
RL	12	4	74%	74%	84%	24.85	9.4	837
SPT	12	4	74%	74%	83%	24.37 ± 0.03	9.01 ± 0.15	837
FIFO	10	4	76%	87%	92%	102.86 ± 2.94	86.38 ± 2.93	857
RL	10	4	71%	69%	81%	22.9	8.37	857
SPT	10	4	71%	69%	81%	22.67 ± 0.02	8.17 ± 0.09	857

In the FIFO scheduling rule, the shuttle agent selects the earliest arriving transaction from its queue. In the SPT rule, the shuttle agent selects the transaction whose tier address is the closest one to its current tier. The simulation models for those scheduling rules are run for 10 independent replications. In the arrival scenarios, a single entity (i.e., transaction) arrival as well as batch (multiple transactions) arrivals at a time are considered. An arriving entity's transaction type (i.e., storage or retrieval) is assigned randomly by assuming equal probability. The simulation results are summarized for averages of 10 replications along with their 95% confidence intervals (CI) in the following tables. Note that since RL-based models do not have multiple replications, the results are shown without CI.

In the simulation models, the mean arrival rates are adjusted so that the bottleneck server's (i.e., Lift 1, Lift 2, or shuttle) average utilization is larger than 90%. Since the FIFO scheduling rule produces the highest results for the average cycle time per transaction performance metric, we first experiment and fix the mean arrival rate for that scenario. Then, to make fair comparisons, we run the other rule scenarios at the same rate. Table 3 summarizes the results along with the considered arrival rates. As mentioned in the model assumptions section, arrivals of transactions follow a Poisson process.

From Table 3, it is observed that the RL and SPT scheduling results are very close to each other and they outperform the FIFO rule. The reason for the closeness of the RL and SPT rules would be that, in the RL approach, we consider the flow time as the reward return for a state-action pair. Thus, by the Q -matrix, the RL agent tends to select the transaction that would result in the

Table 4
Experimental results for batch transaction arrivals

Rule	T	N_s	UL_{avg}	ULS_{avg}	US_{avg}	C_{avg} (seconds)	W_{avg} (seconds)	W (seconds)	Arrival rate (transaction/10 minute)
FIFO	15	5	87%	93%	91%	306.16 ± 5.12	287.5	76	158
RL	15	5	58%	13%	57%	178.98	166.17	44	158
SPT	15	5	58%	13%	58%	188.96 ± 2.56	176.01	47	158
FIFO	15	3	67%	79%	91%	299.14 ± 4.78	281.7	47	100
RL	15	3	45%	15%	56%	173.05	161.7	27	100
SPT	15	3	45%	14%	56%	177.31 ± 2.69	165.9	28	100
FIFO	12	4	79%	90%	93%	384.83 ± 4.86	367.75	86	140
RL	12	4	52%	10%	57%	179.06	168.04	39	140
SPT	12	4	52%	10%	57%	183.93 ± 2.07	172.79	40	140
FIFO	10	4	77%	88%	92%	399.74 ± 5.08	383.23	94	143
RL	10	4	52%	8%	56%	175.98	165.54	40	143
SPT	10	4	51%	8%	56%	180.18 ± 2.47	169.6	41	143

least flow time (i.e., travel time) as in the SPT rule. Also, because in the state definition, we do not consider the bay location addresses as well as the number of tasks waiting in queues, and so forth, the environment might not be captured completely, and the RL agent might behave as if an SPT rule. Also, because, as in the SPT rule, the RL approach considers the tier addresses as actions, the RL results tend to be close to the SPT results. Namely, although a transaction might be at a longer tier travel, it might have the least travel time due to its bay location. By ignoring the bay locations in the action definition, the RL agent might skip such better selections. Hence, it tends to behave as if the SPT rule is considered. Note that including the bay addresses in the state definition is suggested as future work in the conclusion section.

Note that when a shuttle becomes available, it selects a transaction from the shuttle queue based on the Q -matrix. If there were few transactions waiting, the RL agent would not have many options to evaluate. In order to let the RL agent evaluate more waiting transaction options in action selection and observe how the RL approach behaves under the transaction's long queue waiting, we consider batch arrivals in the second arrival scenario case. Table 4 summarizes the batch arrival scenarios results. Note that the mean arrival rates in Table 4 are derived from Table 3 rates by creating batch arrivals every 10 minutes in the simulation models.

From Table 4, it is again observed that both SPT and RL-based scheduling approaches outperform the FIFO scheduling rule. This time, in the batch arrival scenario, the RL approach produces better results than the SPT. This is probably because in action selection, the RL agent can evaluate more transaction options and select the best option according to its goal intelligently.

Last, we ignore the FIFO rule and focus on the comparison of SPT- and RL-based scheduling approaches. In the batch arrival case, remember that we observe that when there are more waiting transactions in action evaluation, the RL-based approach tends to produce better results. Hence, this time, we increase the mean arrival rate scenarios so that we can obtain large (i.e., larger than 85%) utilization values in the SPT and RL approaches. We run the simulation models for batch transaction arrival scenarios under those arrival rates. The results are summarized in Table 5.

Table 5

Experimental results for batch transaction arrival for SPT and RL approaches under higher arrival rates

Rule	T	N_s	UL_{avg}	ULS_{avg}	US_{avg}	C_{avg}	W_{avg}	W	Arrival rate (transaction/minute)
RL	15	5	87%	21%	86%	175.75	162.76	64	150/6.4 minute
SPT	15	5	86%	20%	87%	181.82 ± 1.56	168.7	66	150/6.4 minute
RL	15	3	73%	17%	86%	249.87	238.82	63	150/9.5 minute
SPT	15	3	69%	15%	85%	252.56 ± 2.36	241.5	64	150/9.5 minute
RL	12	4	81%	19%	86%	160.81	149.58	52	120/5.8 minute
SPT	12	4	78%	18%	86%	161.52 ± 0.94	150.3	52	120/5.8 minute
RL	10	4	74%	12%	87%	126.62	116.07	41	100/4.7 minute
SPT	10	4	77%	17%	85%	132.97 ± 1.12	122	43	100/4.7 minute

From Table 5, it is observed that in the batch arrival case and at higher utilization of servers, the RL approach still produces better results than the SPT approach.

5. Conclusion and future work

This paper studies an ML (i.e., Q -learning) method application for SBS/RS for intelligent transaction selection of shuttle agents to process transactions with decreased flow time. Specifically, our aim is to implement an adaptive and dynamic scheduling rule that is developed on real-time state-action information by applying an RL algorithm. The RL algorithm is a “model-free” approach in which an intelligent RL agent is trained on its environment by trial and error. The advantage of this approach is that it is adaptive, and it is able to overcome today’s major challenges of complex industry systems by finding highly complex and non-linear patterns in environmental data. They can be later applied for prediction, detection, classification, regression, or forecasting. By applying ML-based modeling to an automated warehousing problem, this paper aims to be a pioneer in the field of warehouse management implementing AI methods. We simulate the proposed system by using the ARENA 16.0 commercial software by integrating the RL codes in the VBA interface.

We compare the RL-based scheduling approach with FIFO and SPT scheduling rules under different warehouse designs. It is observed that RL outperforms the FIFO scheduling rule. Although the RL produces close values to the considered SPT results, when we consider batch arrivals, the RL results produce slightly better results than the SPT results. Hence, this work provides promising results; if we utilize a more complex deep Q -learning approach than a basic Q -learning approach, we might obtain better results than SPT.

In this paper, we show that RL is applicable for a transaction selection problem in an automated warehousing system. The RL method can also be implemented for any job selection problem in industries. As a future work, we suggest implementing a deep learning method to capture more detailed information from the environment. Hence, the results might outperform even SPT. In addition, a sensitivity analysis for different parameter values and computational time comparisons can also be completed.

Acknowledgments

This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) and Slovenian Research Agency: ARRS (Grant Number: 118M180).

References

- Allied Market Research 2020. Automated Storage and Retrieval System Market. Available at: <https://www.alliedmarketresearch.com/automated-storage-and-retrieval-system-market-A06282> (accessed 20 March 2021).
- Arslan, B., Ekren, B.Y., 2021. Smart transaction picking in tier-to-tier SBS/RS by Deep Q-learning. 11th Annual International Conference on Industrial Engineering and Operations Management, March 7–11, Singapore, pp. 6415–6425.
- Bengio, Y., Lodi, A., Prouvost, A., 2021. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research* 290, 2, 405–421.
- Bertolini, M., Mezzogori, D., Neroni, M., Zammori, F., 2021. Machine learning for industrial applications: a comprehensive literature review. *Expert Systems with Applications* 175, 114820.
- Carlo, H., Vis, I., 2012. Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics* 140, 844–853.
- Dematic (n.d.). Dematic Multishuttle. Available at: <https://www.dematic.com/en/products/products-overview/storage-systems/dematic-multishuttle/> (accessed 20 March 2021).
- Dou, J., Chen, C., Yang, P., 2015. Genetic scheduling and reinforcement learning in multirobot systems for intelligent warehouses. *Mathematical Problems in Engineering* 2015, 597956.
- Eder, M., 2019. An analytical approach for a performance calculation of shuttle-based storage and retrieval systems. *Production and Manufacturing Research* 7, 1, 255–270.
- Ekren, B.Y., 2017. Graph-based solution for performance evaluation of shuttle-based storage and retrieval system. *International Journal of Production Research* 55, 21, 6516–6526.
- Ekren, B.Y., 2020a. A simulation-based experimental design for SBS/RS warehouse design by considering energy related performance metrics. *Simulation Modelling Practice and Theory* 98, 101991.
- Ekren, B.Y., 2020b. A multi-objective optimisation study for the design of an AVS/RS warehouse. *International Journal of Production Research* 59, 4, 1107–1126.
- Ekren, B.Y., Akpunar, A., 2021. An open queueing network-based tool for performance estimations in a shuttle-based storage and retrieval system. *Applied Mathematical Modelling* 89, 1678–1695.
- Ekren, B.Y., Akpunar, A., Sari, Z., Lerher, T., 2018. A tool for time, variance and energy related performance estimations in a shuttle-based storage and retrieval system. *Applied Mathematical Modelling* 63, 109–127.
- Ekren, B.Y., Heragu, S., Krishnamurthy, A., Malmberg, C., 2013. An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. *IEEE Transactions on Automation Science and Engineering* 10, 205–215.
- Ekren, B.Y., Heragu, S.S., 2010a. Approximate analysis of load-dependent generally distributed queueing networks with low service time variability. *European Journal of Operational Research* 205, 2, 381–389.
- Ekren, B.Y., Heragu, S.S., 2011. Simulation based performance analysis of an autonomous vehicle storage and retrieval system. *Simulation Modelling Practice and Theory* 19, 7, 1640–1650.
- Ekren, B.Y., Heragu, S.S., 2012. Performance comparison of two material handling systems: AVS/RS and CBAS/RS. *International Journal of Production Research* 50, 15, 4061–4074.
- Ekren, B.Y., Heragu, S.S., Krishnamurthy, A., Malmberg, C.J., 2010. Simulation-based experimental design to identify factors affecting performance of AVS/RS. *Computers and Industrial Engineering* 58, 1, 175–185.
- Ekren, B.Y., Heragu, S.S., Krishnamurthy, A., Malmberg, C.J., 2014. Matrix-geometric solution for semi-open queueing network model of autonomous vehicle storage and retrieval system. *Computers and Industrial Engineering* 68, 1, 78–86.
- Ekren, B.Y., Sari, Z., Lerher, T., 2015. Warehouse design under class-based storage policy of shuttle-based storage and retrieval system. *IFAC-PapersOnLine*, 28, 3, 1152–1154.

- Emilio, M., Elena, T., Veronique, L. et al., 2021. Exploring the application of machine learning to the assembly line feeding problem. *Operations Management Research* 14, 403–419.
- Eroglu, E., Ekren, B.Y., 2022. An agent-based simulation model for deadlock prevention in an aisle-to-aisle SBS/RS. In Calisir, F. (ed.) *Industrial Engineering in the Internet of Things World*. Cham: Springer International Publishing, pp. 135–146.
- FEM. 2017. Statistics for S/R Machines Results. Available at: <https://www.fem-eur.com/wp-content/uploads/2019/02/IS-Annual-Factsheet-2017.pdf> (accessed 20 March 2021).
- Gosavi, A., 2015. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Boston: Springer.
- Ha, Y., Chae, J., 2018a. Free balancing for a shuttle-based storage and retrieval system. *Simulation Modelling Practice and Theory* 82, 12–31.
- Ha, Y., Chae, J., 2018b. A decision model to determine the number of shuttles in a tier-to-tier SBS/RS. *International Journal of Production Research* 57, 1–22.
- Heragu, S.S., Cai, X., Krishnamurthy, A., Malmberg, C.J., 2011. Analytical models for analysis of automated warehouse material handling systems. *International Journal of Production Research* 49, 22, 6833–6861.
- Jennings, N., 1996. Coordination techniques for distributed artificial intelligence. *Foundations of Distributed Artificial Intelligence* 187–210.
- Kazemi, M., Asef-Vaziri, A., Shojaei, T., 2019. Concurrent optimization of shared location assignment and storage/retrieval scheduling in multi-shuttle automated storage and retrieval systems. *IFAC-PapersOnLine* 52, 13, 2531–2536.
- Kucukyasar, M., Ekren, B.Y., 2022. Two scheduling rule comparisons for operations of shuttles in tier-to-tier SBS/RS. In Calisir, F. (ed.) *Industrial Engineering in the Internet of Things World*. Cham: Springer International Publishing, pp. 91–100.
- Kucukyasar, M., Ekren, B.Y., 2022a. An experimental design study to identify significant factors affecting tier-to-tier SBS/RS performance. In Calisir, F. (ed.) *Industrial Engineering in the Internet of Things World*. Cham: Springer International Publishing, pp. 191–201.
- Küçükyaşar, M., Ekren, B., Lerher, T., 2020. Cost and performance comparison for tier-captive and tier-to-tier SBS/RS warehouse configurations. *International Transactions in Operational Research* 28, 4, 1847–1863.
- Küçükyaşar, M., Ekren, B.Y., Lerher, T., 2021. Energy efficient automated warehouse design. In Vacca, J.R. (ed.) *Solving Urban Infrastructure Problems Using Smart City Technologies*. Amsterdam, the Netherlands: Elsevier, pp. 269–292.
- Lerher, T., 2015. Travel time model for double-deep shuttle-based storage and retrieval systems. *International Journal of Production Research* 54, 1–22.
- Lerher, T., Ekren, B., Dukic, G., Rosi, B., 2015a. Travel time model for shuttle-based storage and retrieval systems. *The International Journal of Advanced Manufacturing Technology* 78, 1705–1725.
- Lerher, T., Ekren, B.Y., Sari, Z., Rosi, B., 2015b. Simulation analysis of shuttle-based storage and retrieval systems. *International Journal of Simulation Modelling* 14, 1, 48–59.
- Lerher, T., Ekren, B.Y., Sari, Z., Rosi, B., 2016. Method for evaluating the throughput performance of shuttle-based storage and retrieval systems. *Tehnicki Vjesnik-Technical Gazette* 23, 715–723.
- Liu, Z., Wang, Y., Jin, M., Wu, H., Dong, W., 2021. Energy consumption model for shuttle-based Storage and Retrieval Systems. *Journal of Cleaner Production* 282, 124480.
- Malus, A., Kozjek, D., Vrabčić, R., 2020. Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning. *CIRP Annals* 69, 1, 397–400.
- Marchet, G., Melacini, M., Perotti, S., Tappia, E., 2011. Analytical model to estimate performances of autonomous vehicle storage and retrieval systems for product totes. *International Journal of Production Research* 50, 12, 7134–7148.
- Marchet, G., Melacini, M., Perotti, S., Tappia, E., 2013. Development of a framework for the design of autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 51, 14, 4365–4387.
- ReportLinker. 2020. Automated Material Handling (AMH) Market–Growth, Trends, and Forecast (2020–2025). Available at: https://www.reportlinker.com/p05815026/Automated-Material-Handling-AMH-Market-Growth-Trends-and-Forecast.html?utm_source=GNW (accessed 20 March 2021).

- Sandholm, T., 2002. An algorithm for optimal winner determination in combinatorial auctions. *IJCAI International Joint Conference on Artificial Intelligence*, 1, 542–547.
- Shyalika, C., Silva, T. & Karunananda, A., 2020. Reinforcement learning in dynamic task scheduling: a review. *SN Computer Science* 1, 306.
- Sutton, R.S., Barto, A.G., 2015. *An Introduction to Reinforcement Learning*. Cambridge, MA: MIT Press.
- Takahashi, K., Tomah, S., 2020. Online optimization of AGV transport systems using deep reinforcement learning. *Bulletin of Networking, Computing, Systems, and Software* 9, 1, 53–57.
- Tappia, E., Roy, D., de Koster, R., Melacini, M., 2016. Modeling, analysis, and design insights for shuttle-based compact storage systems. *Transportation Science* 51, 1, 269–295.
- Wang, Y., Mou, S., Wu, Y., 2015. Task scheduling for multi-tier shuttle warehousing systems. *International Journal of Production Research* 53, 19, 5884–5895.
- Watanabe, M., Furukawa, M., Kakazu, Y., 2001. Intelligent AGV driving toward an autonomous decentralized manufacturing system. *Robotics and Computer-Integrated Manufacturing* 17, 1–2, 57–64.
- Wauters, T., Villa, F., Christiaens, J., Alvarez-Valdes, R., Vanden Berghe, G., 2016. A decomposition approach to dual shuttle automated storage and retrieval systems. *Computers and Industrial Engineering* 101, 325–337.
- Xue, T., Zeng, P., Yu, H., 2018. A reinforcement learning method for multi-AGV scheduling in manufacturing. 2018 IEEE International Conference on Industrial Technology (ICIT), February 20–22, Lyons, France, pp. 1557–1561.
- Zhao, X., Wang, Y., Wang, Y., Huang, K., 2019. Integer programming scheduling model for tier-to-tier shuttle-based storage and retrieval systems. *Processes* 7, 4, 223.
- Zou, B., Xu, X., Gong, Y., De Koster, R. 2016. Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems. *European Journal of Operational Research* 254, 1, 51–67.