PLOS ONE

RESEARCH ARTICLE

Modeling and performance analysis of shuttle-based compact storage systems under parallel processing policy

Lei Deng[1]*, **Lei Chen**[1], **Jingjie Zhao**[2], **Ruimei Wang**[3]

1 School of Information, Beijing Wuzi University, Beijing, China, **2** Beijing Municipal Tax Service, State Taxation Administration, Beijing, China, **3** College of Economics and Management, China Agricultural University, Beijing, China

* dengleibwu@126.com

Citation: Deng L, Chen L, Zhao J, Wang R (2021) Modeling and performance analysis of shuttle-based compact storage systems under parallel processing policy. PLoS ONE 16(11): e0259773. https://doi.org/10.1371/journal.pone.0259773

Editor: Behzad Behdani, Wageningen University, NETHERLANDS

Received: May 27, 2021

Accepted: October 27, 2021

Published: November 15, 2021

Data Availability Statement: All relevant data are within the manuscript and its Supporting information files.

Funding: This work was supported by Beijing Social Science Foundation (No. 19GLC043). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Short response time for order processing is important for modern warehouses, which can be potentially achieved by adopting appropriate processing policy. The parallel processing policy have advantages in improving performance of many autonomous storage and retrieval systems. However, researchers tend to assume a sequential processing policy managing the movement of independent resources in shuttle-based compact storage systems. This paper models and analyses a single-tier of specialized shuttle-based compact storage systems under parallel processing policy. The system is modeled as a semi-open queueing network with class switching and the parallel movement of shuttles and the transfer car is modeled using a fork-join queueing network. The analytical model is validated against simulations and the results show our model can accurately estimate the system performance. Numerical experiments and a real case are carried out to compare the performance of parallel and sequential processing policies. The results suggest a critical transaction arrival rate and depth/width ratio, below which the sequential processing policy outperforms the parallel processing policy. However, the advantage of sequential processing policy is decreasing with the increasing of shuttle number, transaction arrival rate and depth/width ratio. The results also suggest an optimal depth/width ratio with a value of 1.75 for minimizing the expected throughput time in the real system. Given the current system configurations, the parallel processing policy should be considered when the number of shuttles is larger than 2 or the transaction arrival rate is larger than 24 per hour.

1 Introduction

In recent years, customer demands for logistics and distribution become dynamic and keep changing, especially during COVID-19. This implies an increasing trend towards more service variety and shorter response times. As a new unit-load storage and retrieval system, shuttle-based compact storage systems combine the features of and are more cost-effective than autonomous vehicle-based storage systems and compact storage systems. Additionally, such systems

are more time-saving and flexible, which means they have a shorter response time for storage or retrieval transactions and can change their throughput capacity by adding or removing shuttles [1, 2]. All these potential advantages of shuttle-based compact storage systems result to its growing popularity among and higher adoption by modern warehouses [3]. The shuttle-based compact storage systems consist of multiple tiers of multi-deep storage lanes. In such systems, the vertical movements moving loads across tiers are carried out using lifts and horizontal movements moving loads within the storage lanes are carried out using shuttles [1, 4]. The horizontal movements of a shuttle within cross-aisle, which is orthogonal to the storage lanes, can be performed either by a transfer car or by the shuttle itself. The shuttle that is transported to and from appropriate storage lanes by the transfer car is called specialized shuttle (Fig 1), while the one that can move both within the storage lanes and along cross-aisle is called generic shuttle. As our purpose is to model parallel movements of shuttles and transfer car, we only consider the systems with specialized shuttles.

In practice, the autonomous storage and retrieval systems are widely used and studied. Most of the literatures on this subject focus on the autonomous vehicle-based storage and retrieval systems (AVS/RS). In such systems, the vertical movements are performed by vehicles and horizontal movements are carried out by lifts. The most studied systems are characterized by multiple tiers of single- or double-deep storage racks. Malmborg [5] is the first to study AVS/RS systems. He builds a continuous markov chain model to describe the characters of the system and use a state equation model to estimate the utilization of servers and cycle time of system with the consideration of both single- and dual-command cycle times. Based on this

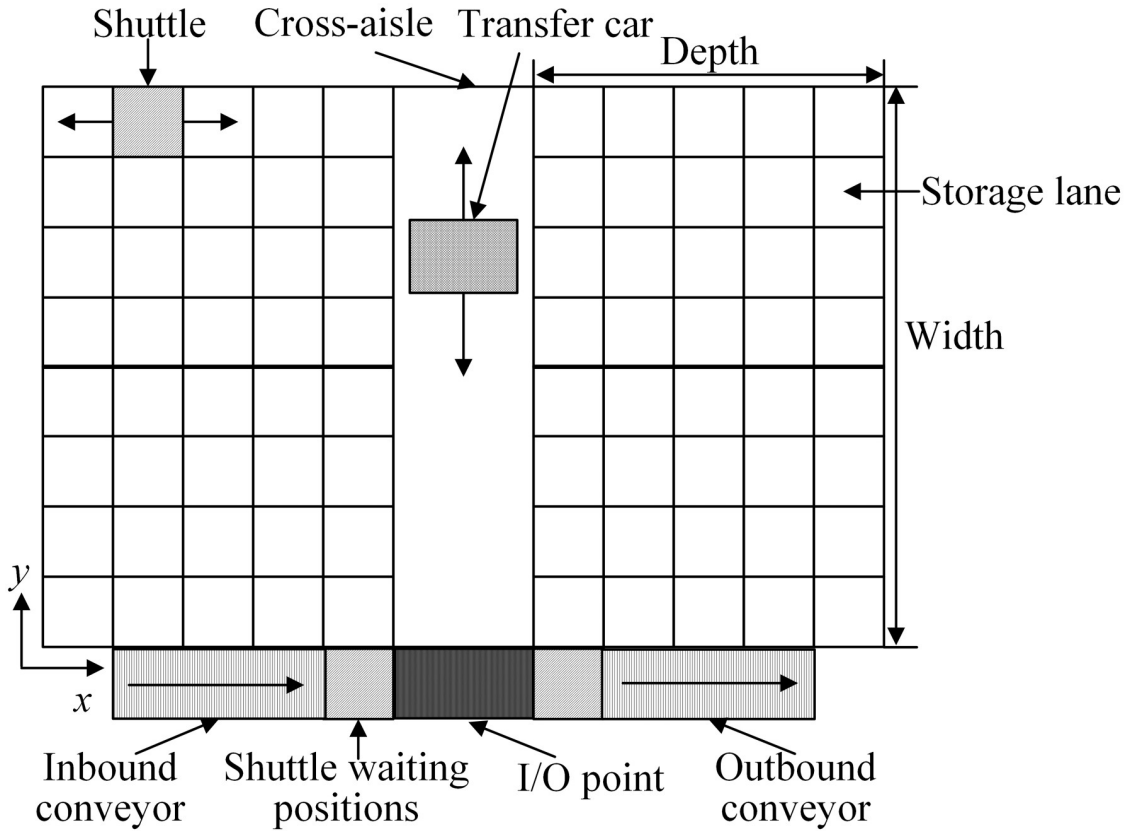

Fig 1. Top view of the shuttle-based compact storage system.

https://doi.org/10.1371/journal.pone.0259773.g001

work, Malmborg [6] attempts to propose an optimal system design through comparing AVS/RS systems to traditional autonomous storage systems, and his results shows the former one has advantages in cost-savings and operational flexibility. And Malmborg [7] extends the previous works by considering the opportunistic-interleaving in the system. By using queueing method, the activity of vehicle is modeled by an M/G/V queue and the activity of lifts are modeled by a G/G/L queue in the research of Fukunari and Malmborg [8]; Kuo et al. [9]. Roy et al. [10] analyze a single-tier AVS/RS using semi-open queueing network and examines the system performance. Furthermore, as a useful modelling tool, semi-open queueing network is used to model AVS/RS by the works of Heragu et al. [11] and Marchet et al. [12]. Nevertheless, there are many researches analyze AVS/RS through simulation, such as Ekren [4]. Recently, Ekren and Akpunar [13] develop an open queuing network and a software-based tool to calculate the performance of AVS/RS. In this research, they consider both single- and dual-command cycles, and also estimate system performance related to energy consumption.

For the shuttle-based storage and retrieval systems (SBS/RS), some studies focus on the system design, energy consumption and scheduling process. For instance, Zhao et al. [14] use a semi-open queueing network (SOQN) to model a tier-to-tier SBS/RS system to identify the optimal number of shuttles and provide some insights in system design. Through simulation analysis, Ha and Chae [15] propose a free balancing in SBS/RS systems to prevent collisions and blockages and achieve the targeted system throughput with an optimal number of shuttles. Wu et al. [16] build a queueing model and design an optimal algorithm to find the minimum cost configurations in terms of number of tiers, aisles, lifts and workstations with given throughput, tote capacity and order cycle time requirements. Lei et al. [17] investigate the optimal storage location assignment by using a optimization model. Besides, Luo et al. [18] and Dong et al. [19] investigate the optimal scheduling rule for storage and retrieval processes, respectively, to minimize the makespan of storing or retrieving a series of loads. And Liu et al. [20] develops an energy consumption model for the SBS/RS and estimate the maximum energy consumption under different throughput requirement.

Studies on the shuttle-based compact storage systems are scant, notwithstanding its better volume flexibility, lower operational cost and shorter respond time. Tappia et al. [2] consider multiple tiers and build a semi-open queueing network to model this system. Based on the results of their analytical models, they show the optimal depth/width ratio and number of tiers and compare the economic performance between specialized and generic shuttles. Compared to the research of Tappia et al. [2], Manzini et al. [21] only focus on the estimation of travel time and distance, aiming to find an appropriate layout and system configuration to optimize the system performance in terms of travel distance and cycle time. Borovinšek et al. [3] attempt to find out the optimal layout and system configuration to minimize the investment, energy consumption and cycle time of the system by using a multi-objective optimization model. D'Antonio et al. [22] consider the effect of different allocation criterion on system performance and propose an analytical model based on probabilistic approach to estimate the cycle time and its standard deviation. Boysen et al. [23] focus on a shuttle-based deep-lane storage system with forklifts performing vertical movements. They build a mixed-integer programming model to estimate the performance of two system configurations, namely one-sided and two-sided access to deep-lane storage system, aiming at avoiding blocking. Eder [24] proposes a continuous-time open queueing network taking into account the effect of capacity limitation and the results show that as the increasing of storage depth, the throughput time increases and the investment cost decreases. Recently, Kumawat and Roy [25] develop a new solution approach to solve the multi-stage semi-open queuing networks and apply it in the shuttle-based compact storage systems, which is more accurate for estimating system performance.

Our literature review shows that a sequential processing policy is used to manage the movement of shuttles and transfer car. For instance, when a retrieval transaction is assigned to a shuttle, the shuttle travels to the first bay of its lane and place a request for transfer car sequentially. Once the transfer car is available, it moves to shuttle's lane, transports the shuttle to the retrieval lane, releases the shuttle, waits for shuttle retrieving the load and then transports the shuttle to the I/O point. During the retrieval transaction, the transfer car cannot respond for demand of any other shuttles, which means the longer the time that the shuttle retrievals load takes, the more inefficient the whole system will be. As pointed out by Tappia et al. [2], the sequential processing policy is currently in use for some warehouses since their storage lanes are not too deep. As the storage lanes become deeper, however, it will take more time for transfer car in waiting for shuttles retrieving loads. Under the parallel processing policy, the movements of shuttles within storage lanes and transfer car in the cross-aisle are simultaneous (Fig 1). Some previous studies have examined the performance of such a policy in automated and vehicle-based storage and retrieval systems [26, 27]. The systems in their studies are crane-based [26] or single/double deep storage systems [27], which are differ from the shuttle-based compact storage systems discussed in our study. Besides, the former research uses deterministic models and the latter only takes retrieval transactions into consideration. Recently, Kumawat et al. [28] propose a closed queueing network with two-phase servers to model the simultaneously operations of shuttle and transfer car in a shuttle-based compact storage system. However, their model only captures the parallel movements of shuttles and transfer car before their joint movement, meaning the transfer car still has to wait for shuttle moving within storage lane to pick up the load.

In summary, simultaneously operations of independent resources in autonomous storage and retrieval systems have attracted the attention of scholars who have performed a number of theoretical studies. However, existing studies on shuttle-based compact storage systems either assume sequential operations between shuttle and transfer car or focus on the modeling of parallel movements of shuttle and transfer car before their joint movement in cross-aisle, both of which mean that the transfer car has to wait for shuttle retrieving the load. In practice, the simultaneous movements of different resources when processing a transaction have advantages in system performance over their sequential movements [28]. Despite the requirement for shorter response times and the performance benefits of parallel processing policy, most previous literatures mainly focus on the sequential processing policy and studies on the parallel processing policy are rare.

Therefore, to contribute to the scant literature on this subject, this study aims to estimate the system performance under parallel processing policy and investigate the conditions on which the parallel processing policy outperforms the sequential processing policy. Based on this, this study analyzes the operational processes of shuttle-based compact storage systems under parallel processing policy and develop a multi-class semi-open queuing network (SOQN) with class switching to model such system. Meanwhile, a fork-join queueing network (FJQN) is used to model the concurrent movement of shuttles and transfer car. Since the original network does not have a product-form solution, a decomposition-based approximation approach is developed to estimate the system performance and simulation is used to validate the accuracy of the analytical model. Additionally, a series of numerical experiments are conducted to compare the system performance under parallel and sequential processing policies. Some design insights and managerial implications are provided through the investigation of a real case. With respect to the previous literature, this study mainly focuses on the parallel movements of shuttles and the transfer car. The results may provide new insights for the improvement of warehouse performance. The main contributions of this study are the followings:

1. We develop a SOQN combined with FJQN to model the parallel movements of shuttles and the transfer car in shuttle-based compact systems. Compared to the previous studies, our model is stochastic and considers both storage and retrieval transactions, thereby taking into account the effect of time spent on waiting for resources to be paired and the route of shuttles. Besides, our model allows the transfer car to be released and respond for the demand of another shuttles when the shuttle is retrieving the load (in the existing studies, the transfer car have to wait for shuttle moving within storage lane to pick up the load).
2. We validate the proposed model using numerical experiments and apply the model on a real case and compare it with the model under sequential processing policy proposed by Tappia et al. [2]. Our analytical results provide some managerial insights in regards to the conditions in terms of number of shuttles, depth/width ratio and arrival rate of orders, under which the parallel processing policy should be considered.

The rest of the study is organized as follows: section 2 provides the system description and assumptions. In section 3, we introduce the models and the approximate solution approach is described in section 4. Section 5 contains the simulation validation, numerical experiments and the insights. Conclusions and future works are presented in section 6.

2 System description and assumptions

2.1 Main notations and assumptions

Table 1 summarized main notations used throughout the study.

The following assumptions are made in this study:

Table 1. Main notations.

Notation	Description
λ_r, λ_s	Arrival rate of retrieval and storage transactions
N_s	Number of shuttles
N_c, N_l	Number of storage columns and lanes at each side of cross-aisle.
w, d	Unit width and depth per storage position
t_t, t_{sh}	Constant time required for transfer car or shuttle to load/unload the shuttle or unit load
v_t, v_{sh}	Constant velocity of transfer car and shuttle
$t_{sh1}, t_{sh2}, t_{sh3}$	Expected travel time related to shuttles
t_{t1}, t_{t2}, t_{t3}	Expected travel time related to transfer car
p_s, p_r	Probability of storage and retrieval transaction
p_{sin}, p_{sio}	Probability that a transaction is assigned to shuttle dwelling at interior or I/O point
p_{cin}, p_{cio}	Probability that the transfer car dwells at interior or I/O point
p_{ss}, p_{sd}	Probability that the assigned shuttle is or is not present in the lane where the retrieval load is present
T_{ir}, T^c_{iu}	Mean service time of node i for class r customer or chain u customer
e_{ir}, e^c_{iu}	Mean number of visits of a class r customer or chain u customer at node i
$p_{ir,js}$	Probability that a class r customer at the ith node is transferred to class s and the jth node
$p_{0,js}$	Probability that a class s customer from outside enters the jth node
$p_{js,0}$	Probability that a class s customer leaves the system after the service at the jth node
$E[T]$	Expected throughput time of the system
U_{sh}, U_t	Average utilizations of shuttle and transfer car
L_o, L_{sh}	Average number of transactions and free shuttles waiting at external queue of system
L_f, L_i	Mean queue length at fork-join node and at node i
dw	The depth/width ratio of system

https://doi.org/10.1371/journal.pone.0259773.t001

1. We only consider a single tier. This is based on the following observations. First, the parallel movements of shuttle and transfer car are performed within a single tier. Second, our model can be easily extended to the case of multiple tiers by using the multi-tier linking approach proposed by Tappia et al. [2].
2. We only consider a system with specialized shuttle, since we are interested in whether the simultaneous operations of shuttle and transfer car improve the system performance.
3. The arrival process of both storage and retrieval transactions are assumed to follow a Poisson distribution.
4. The random storage policy is used, meaning the probability of a product being stored in any storage positions is equal.
5. We consider the storage system operates in single-command cycles, which means only a single storage transaction or a single retrieval transaction is performed and only one unit load is handled in each cycle.
6. Each storage lane holds one product.
7. Since compared with the number of storage lanes, the number of shuttles is small, we assume a storage lane can be accessed by at most one shuttle once so that we can ignore the shuttle blocking effects within a storage lane.
8. The shuttles and the transfer car follow a point-of-service-completion (POSC) dwell point policy. Therefore, the shuttles and the transfer car will wait either at an interior point after completion of a storage transaction or the I/O point after completion of a retrieval transaction.
9. The shuttles and the transfer car follow a first-come-first-served (FCFS) scheduling policy.
10. The arriving transaction is performed by the first available shuttle, or by the first shuttle waiting at the idle shuttle queue regardless of the transaction type and shuttle dwell point.
11. We do not consider the effect of acceleration and deceleration on the movement of shuttles and transfer car.

2.2 System and operational process description

Fig 1 provides a top view of the studied system. A single tier shuttle-based compact storage system with specialized shuttles consists of multiple storage lanes with each lane holding one product. A cross-aisle is located in the middle of the tier, which is orthogonal to the storage lanes. The movement within the storage lanes is performed by shuttles. In the meantime, a transfer car performs the movement along the cross-aisle. There is only one input/output (I/O) point, which is located at the corner of storage lanes and the end of cross-aisle. Shuttles waiting positions are located next to the I/O point. A conveyor moves the loads to be stored from the inbound work station to the shuttle waiting position and the loads to be retrieved from the shuttle waiting position to the outbound work station.

When a transaction is assigned to a shuttle, a request is made by the shuttle for transfer car simultaneously. Given the POSC and FCFS policies, the shuttle and transfer car can dwell at any interior or I/O point and a transaction can be assigned to any shuttle regardless of its dwell point. Besides, whether the shuttle dwells at the same lane of retrieval position results in different individual movements required to perform retrieval transactions. Therefore, depending on the dwell point of shuttle (interior or I/O point) and the type of transaction (storage or

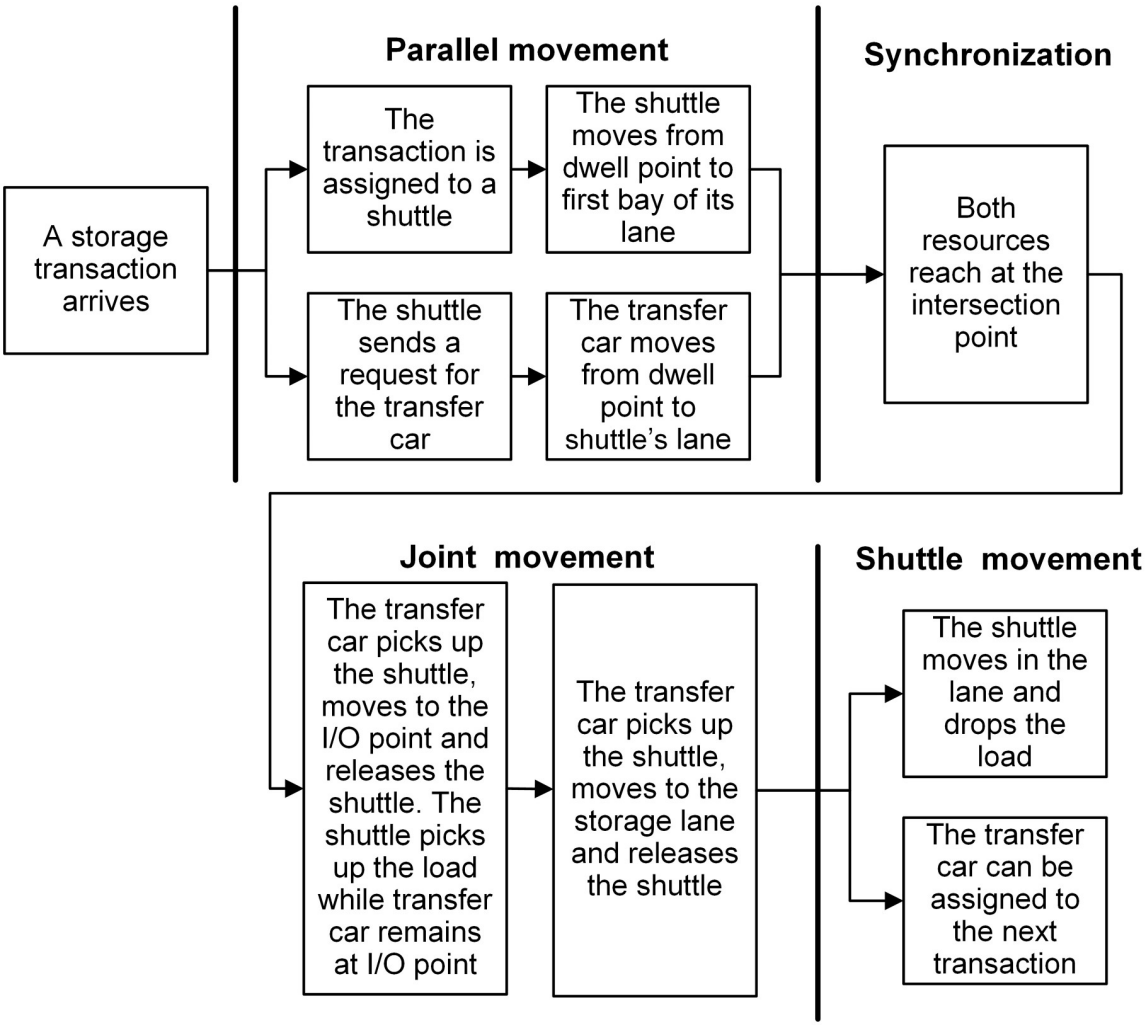

Fig 2. Operational process for storage transactions when the shuttle dwells at an interior point.

https://doi.org/10.1371/journal.pone.0259773.g002

retrieval), one of the following scenarios showed in Figs 2–5 can occur (For details about the operational processes of such a system under sequential processing policy, we refer to Tappia et al. [2]).

2.3 Components of travel time related to shuttles and transfer car

Given the random storage policy and the operational processes showed in Figs 2–5, the expected travel time related to shuttles and transfer car be obtained based on the probability distribution of storing or retrieving a load from each storage position, i.e., a uniform distribution. Therefore, each component of the travel time related to shuttles and transfer car can be expressed as follows:

Time required for the shuttle to:

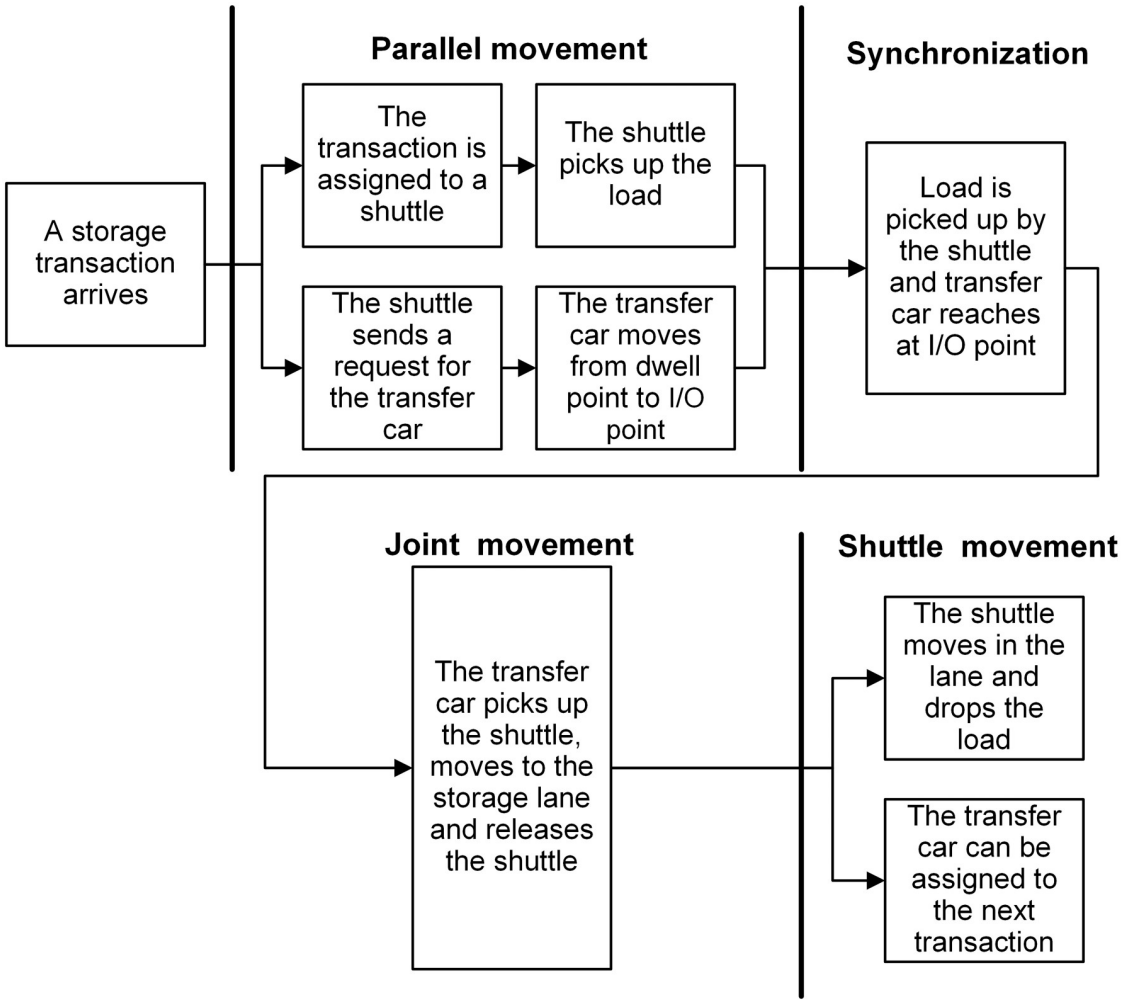

Fig 3. Operational process for storage transactions when the shuttle dwells at the I/O point.

https://doi.org/10.1371/journal.pone.0259773.g003

1. travel from dwell point (or the retrieval position) to the first bay of its lane:

$$t_{sh1} = \sum_{k=1}^{N_c} \frac{1}{N_c} \frac{(k-1)d}{v_{sh}} = \frac{(N_c-1)d}{2v_{sh}} \tag{1}$$

2. travel from dwell point to the retrieval position when dwells in the same lane of retrieval position:

$$t_{sh2} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \frac{1}{N_c^2} \frac{|i-j|d}{v_{sh}} \tag{2}$$

3. pick up or drop the load:

$$t_{sh3} = t_{sh} \tag{3}$$

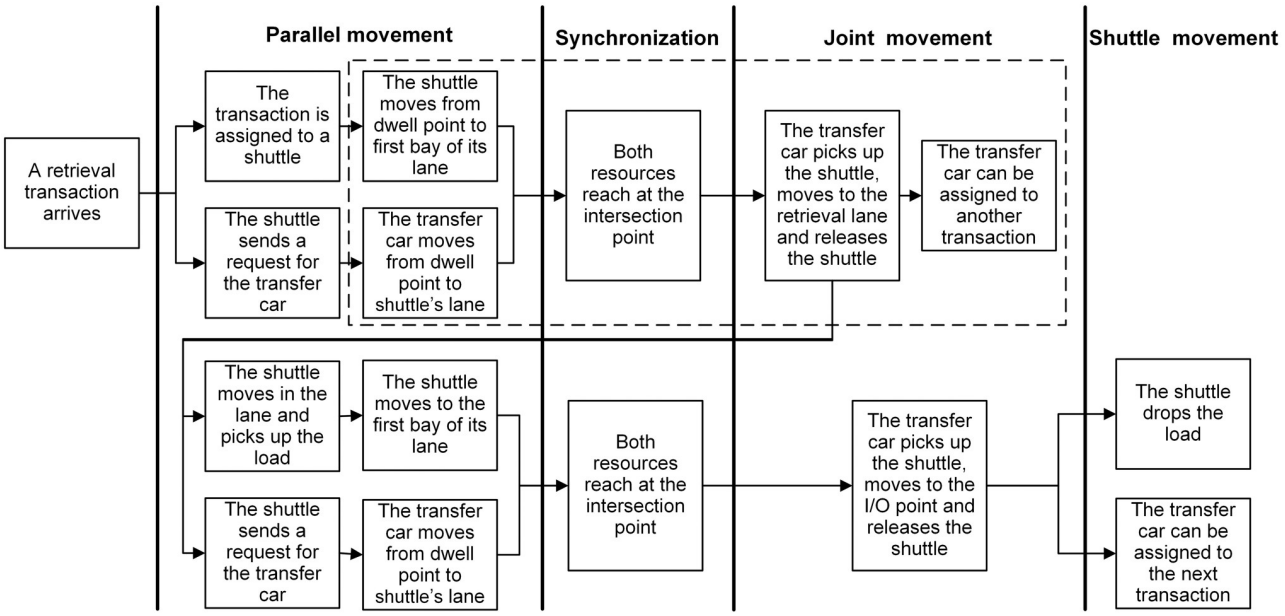

Fig 4. Operational process for retrieval transactions when the shuttle dwells at an interior point. The operational steps within dotted line are performed only when the shuttle is not present in the lane where the retrieval load is present.

https://doi.org/10.1371/journal.pone.0259773.g004

Time required for the transfer car to:

1. travel from I/O point to shuttle's lane or travel from its dwell point (not I/O point) to I/O point:

$$t_{t1} = \frac{N_l w}{2v_c} \tag{4}$$

2. travel from dwell point (not I/O point) to the shuttle's lane:

$$t_{t2} = \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \frac{1}{N_l^2} \frac{|i-j|w}{v_c} \tag{5}$$

3. load or unload the shuttle:

$$t_{t3} = t_t \tag{6}$$

3 Semi-open queueing network for shuttle-based compact storage systems

3.1 Queueing model

Fig 6 shows the SOQN model for the shuttle-based compact storage systems. For modeling purpose, we divide the system operational process into three parts: parallel movement, joint movement and shuttle movement (Figs 2–5). The model considers both storage and retrieval transactions. The shuttles are modeled as resources.

As showed in Fig 6, there are three nodes and a fork-join network with two nodes. All the service required for shuttles and transfer car to complete the parallel movement is modeled by

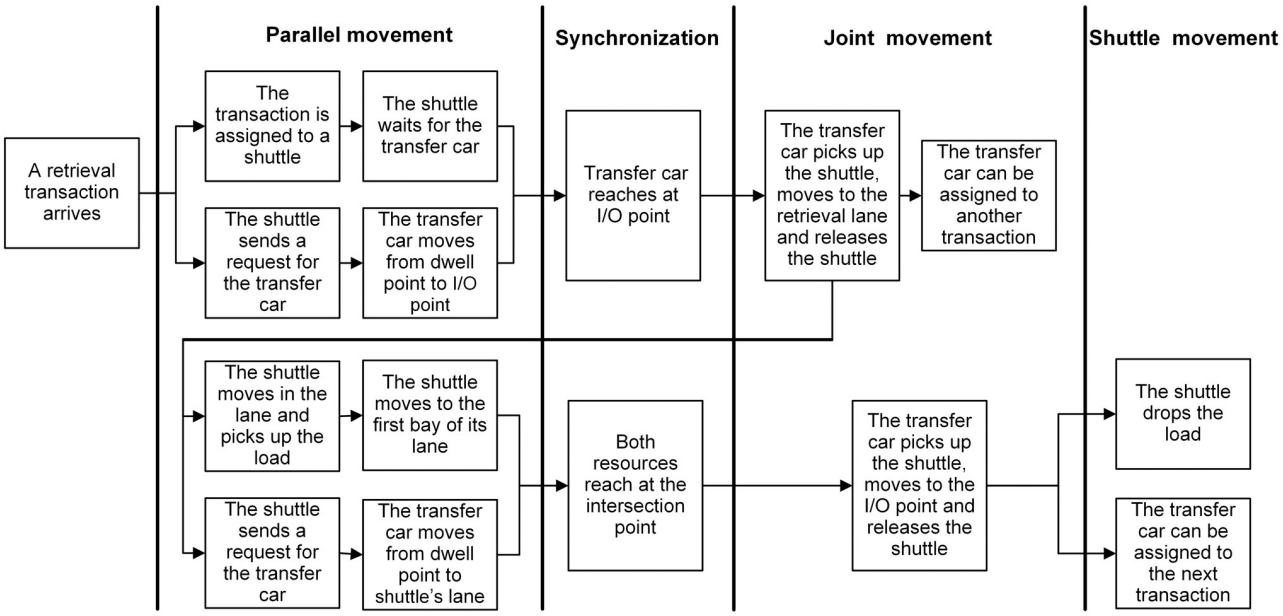

Fig 5. Operational process for retrieval transactions when the shuttle dwells at the I/O point.

https://doi.org/10.1371/journal.pone.0259773.g005

the fork-join network, in which the service of shuttles is represented by an infinite-server (IS) node 1 and the service of transfer car is represented by a single-server node 2. The joint movement is captured by the single-server node 3. IS nodes 4 and 5 represent shuttle movements for retrieval and storage transaction, respectively. Node S is a synchronization station with two queues that Q_1 represents the external queue of transactions and Q_2 represents the queue where shuttles will be released to after completing service. Under the parallel processing policy, an incoming transaction, after being paired with the first available shuttle, is split into two parts that one is served by the shuttle and the other by the transfer car. The completed part waits in one of the two join queues denoted by Q_{sh} (represents the shuttles) and Q_t (represents the transfer car) for the completion of the other one. Then they join at the join node.

Our model allows transfer car performing other tasks after it releases the shuttle to retrieve the required load. This results that the shuttle and transfer car have to be synchronized upon completion of their parallel movements more than once when processing retrieval transactions

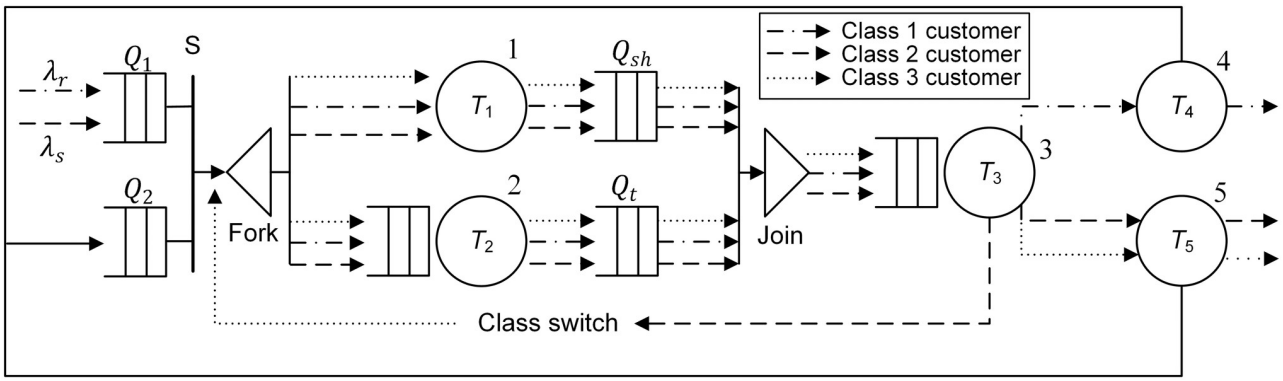

Fig 6. Queueing network model of the shuttle-based compact storage system.

https://doi.org/10.1371/journal.pone.0259773.g006

Table 2. Description of the customer class switching rule.

Customer class prior to the joint movement	Dwell point of shuttle	Same lane	Customer class after the joint movement
1	I/O or interior point		1
2	Interior point	yes	2
2	Interior point	no	3
2	I/O point		3
3			3

Same lane means whether the shuttle is present in the lane where the retrieval load is present.

https://doi.org/10.1371/journal.pone.0259773.t002

(Figs 4 and 5). To deal with the differences between their first and second synchronizations, class switching is allowed in our model. Specifically, we call the transaction the customer of the system, and storage and retrieval transaction as a class 1 and 2 customer, respectively. In the case that a class 2 customer is served by a shuttle dwelling in a storage lane which is different from that of retrieval position, after the transfer car moving to the destination storage lane and unloading the shuttle (the first joint movement which is captured by node 3), the class of the customer changes to 3. While in the case that a class 2 customer is served by a shuttle dwelling at I/O point, after the transfer car moving from I/O point to destination lane and unloading the shuttle, the class of the customer also changes to 3. Table 2 describes the customer class switching rule.

The external arrival process of transactions is assumed to follow a Poisson distribution. Thus, the type of transaction waiting at the head of Q_1 can be storage with probability $p_s = \lambda_s/(\lambda_r + \lambda_s)$ or retrieval with probability $p_r = \lambda_r/(\lambda_r + \lambda_s)$. The probability with which a transaction is assigned to shuttle dwelling at interior (or I/O) point is $p_{sio} = \lambda_r/(\lambda_r + \lambda_s)$ (or $p_{sin} = \lambda_s/(\lambda_r + \lambda_s)$). Since the loads are stored randomly in the system, we can get $p_{ss} = 1/(2N_l)$ and $p_{sd} = (2N_l - 1)/(2N_l)$.

Transfer car dwells at interior or I/O point depends upon the previous task it completes (there are three possible tasks in our model: joint movement of storage transaction and the first and second joint movement of retrieval transaction), which makes it difficult to calculate the corresponding probabilities. Specifically, the transfer car dwells at I/O point after the completion of the first joint movement of retrieval transaction, while it dwells at an interior point after the completion of the joint movement of storage transaction and the second joint movement of retrieval transaction. Given the random storage policy, it is reasonable to assume that, in steady state, the number of class 2 customers (excluding the case that shuttle is present in the lane where the retrieval load is present) is equal to that of class 3 customers. This assumption implies, given that the shuttle is not present in the lane where the retrieval load is present, the probability that the transfer car performs the first joint movement of retrieval transaction is equal to the probability that it performs the second joint movement of retrieval transaction (i.e., both probabilities can be expressed as $(p_r p_{sin} p_{sd} + p_r p_{sio})/2$). Thus, we can obtain:

$$p_{cin} = \frac{1}{2}(p_r p_{sin} p_{sd} + p_r p_{sio}) + p_s = \frac{(2\lambda_r^2 + 6\lambda_r\lambda_s + 4\lambda_s^2)N_l - \lambda_r\lambda_s}{4(\lambda_r + \lambda_s)^2 N_l} \tag{7}$$

$$p_{cio} = \frac{1}{2}(p_r p_{sin} p_{sd} + p_r p_{sio}) + p_r p_{sin} p_{ss} = \frac{(2\lambda_r^2 + 2\lambda_r\lambda_s)N_l + \lambda_r\lambda_s}{4(\lambda_r + \lambda_s)^2 N_l} \tag{8}$$

Table 3. Service time expressions for nodes 3, 4 and 5.

Node	Notation	Mean	Node	Notation	Mean	Probability	Corresponding scenario
4	T_4	$t_{sh1} + t_{sh3}$	3	T_{31}	$2t_{t1} + 2t_{t2} + t_{t3}$	p_{sin}	Fig 2
					$t_{t1} + 2t_{t3}$	p_{sio}	Fig 3
				T_{32}	$t_{t2} + 2t_{t3}$	$p_{sin}p_{sd}$	Fig 4, different lane
5	T_5	t_{sh3}			$t_{t1} + 2t_{t3}$	$p_{sin}p_{ss}$	Fig 4, same lane
					$t_{t1} + 2t_{t3}$	p_{sio}	Fig 5
				T_{33}	$t_{t1} + 2t_{t3}$	1	Fig 5 or different lane in Fig 4

Same or different lane means the shuttle is or is not present in the lane where the retrieval load is present.

https://doi.org/10.1371/journal.pone.0259773.t003

Let f denotes the fork-join node, the routing probabilities are given as follows:

$$p_{0,f1} = p_s, p_{f1,31} = 1, p_{31,41} = 1, p_{41,0} = 1$$
$$p_{0,f2} = p_r, p_{f2,32} = 1, p_{32,52} = p_{sin}p_{ss}, p_{32,f3} = p_{sin}p_{sd} + p_{sio}, p_{52,0} = 1$$
$$p_{f3,33} = 1, p_{33,53} = 1, p_{53,0} = 1$$

3.2 Service time expressions

The service time of each node for each class customer depends upon the type of transactions and the dwell point of shuttles and transfer car. Therefore, based on the scenarios provided in Figs 2–5 and the component of travel times related to shuttles and the transfer car, we calculate the service time expressions for nodes 3, 4 and 5 and summarize in Table 3 and fork-join node in Table 4, as well as their corresponding probabilities and scenarios.

4 Solution approach for semi-open queueing networks

The queueing model we developed is a multiclass semi-open queueing network with both general and infinite stations. It is difficult to evaluate such queueing network directly by continuous-time Markov chain (CTMC) since the system has a large state space. This is because we have to record the number of each customer class in each node and its corresponding queue,

Table 4. Service time expressions for fork-join node.

Scenario	Customer class	Dwell point		Same lane	Probability	Mean service time	
		Shuttle	Transfer car			Node 1	Node 2
1	1	Interior	I/O		$p_s p_{sin} p_{cio}$	t_{sh1}	t_{t1}
2	1	Interior	Interior		$p_s p_{sin} p_{cin}$	t_{sh1}	t_{t2}
3	1	I/O	I/O		$p_s p_{sio} p_{cio}$	t_{sh3}	0
4	1	I/O	Interior		$p_s p_{sio} p_{cin}$	t_{sh3}	t_{t1}
5	2	Interior	I/O	yes	$p_r p_{sin} p_{cio} p_{ss}$	$t_{sh1} + t_{sh2} + t_{sh3}$	t_{t1}
6	2	Interior	Interior	yes	$p_r p_{sin} p_{cin} p_{ss}$	$t_{sh1} + t_{sh2} + t_{sh3}$	t_{t2}
7	2	Interior	I/O	no	$0.5 p_r p_{sin} p_{cio} p_{sd}$	t_{sh1}	t_{t1}
8	2	Interior	Interior	no	$0.5 p_r p_{sin} p_{cin} p_{sd}$	t_{sh1}	t_{t2}
9	2	I/O	I/O		$0.5 p_r p_{sio} p_{cio}$	0	0
10	2	I/O	Interior		$0.5 p_r p_{sio} p_{cin}$	0	t_{t1}
11	3		I/O		$0.5 p_r p_{cio}$	$2t_{sh1} + t_{sh3}$	t_{t1}
12	3		Interior		$0.5 p_r p_{cin}$	$2t_{sh1} + t_{sh3}$	t_{t2}

https://doi.org/10.1371/journal.pone.0259773.t004

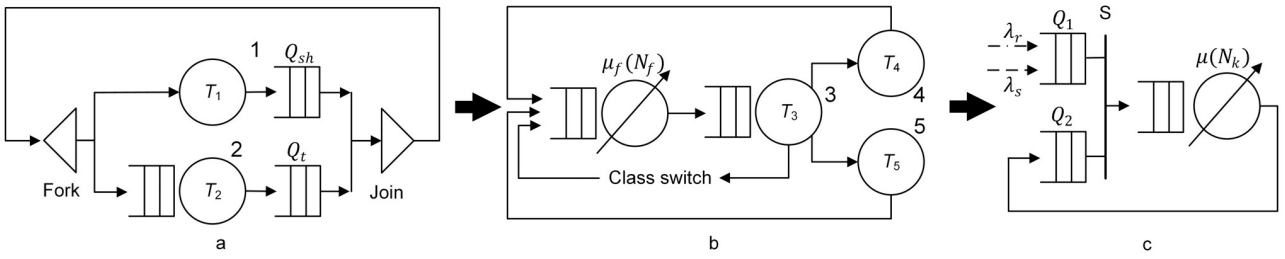

Fig 7. Procedure for reducing the original network.

https://doi.org/10.1371/journal.pone.0259773.g007

the number of idle shuttles and the exact order of all of the customers in join queues. In order to estimate the performance of such model with a non-product form solution, we develop a decomposition-based approximation method including following three steps: first, we consider the FJQN as a closed network and estimate its load-dependent service rate; second, we replace the FJQN by a Flow Equivalent Server (FES), aggregate the compliment network, together with the FES, into a single server and calculate its service rate; at last, we solve the reduced SOQN with one single server directly by CTMC. Fig 7 shows the procedure for reducing the original network to a one single-server network.

4.1 Estimation of load-dependent service rate of the FJQN

There are two single-server stations in the FJQN, one of which is general station representing transfer car and the other is IS representing shuttles (Fig 7a). Note that there is no class switching in this closed queueing network. Thus, to obtain the service rate of FJQN, we first aggregate all classes into one, as suggested by [29], and consider the FJQN as a closed network and short-circuit the other nodes. Thus, we can approximate the state probabilities and calculate the service rate.

The mean service time of node 1 and 2 for the aggregation class is given by the combination of mean service time of all possible scenarios. Therefore, the mean service time of node1 and 2, as well as their second moments are obtained by:

$$E(T_i) = \sum\nolimits_m p_m T_{im} \tag{9}$$

$$E(T_i^2) = \sum\nolimits_m p_m T_{im}^2 \tag{10}$$

where $E(T_i)$ denotes the mean service time of node i, T_{im} represents the mean service time of node i in mth scenario described in Table 3 with its corresponding probability denoted by p_m and $E(T_i^2)$ represents the second moment of expected service time for the aggregation class. Moreover, the squared coefficient of variation (scv) of the service time can be obtained by $cv_i^2 = \left(E(T_i^2) - E(T_i)^2\right)/E(T_i)^2$.

Since the service time of two nodes are general distributed and $cv_i^2 < 1$, an Erlang-k distribution is adopted to approximate the service process of each node, where k denotes the number of exponential phases and $k = [1/cv^2]$. The mean service time at each phase, $\mu^{-1} = E(T)/k$.

When a transaction goes through the FJQN, it will be split into two parts, one of which requests the service of shuttles at node 1, the other requests the service of transfer car at node 2. The joining of these two tasks at the join node of FJQN represents the service completion of a transaction. Thus, the state of the system can be described by a two-dimensional vector $st_q = (N_{ws}, N_{wt})$, where N_{ws} is the number of waiting shuttles in the join queue Q_{sh} and N_{wt} is the

number of waiting transfer cars in the join queue Q_t. Let N_f be the number of customers in the closed network, q be the total number of states in this network, we can obtain $q = N_{ws}(N_f + 1) + N_{wt}$. Given the fact that there is only one transfer car in the system, the state space can be expressed by:

$$N_{ws} + N_{wt} < N_f + 1,\ N_{ws} = 0, 1, \cdots N_f,\ N_{wt} = 0, 1 \tag{11}$$

As shown in [27], the joining of two tasks can be completed when a shuttle finishes its service if $N_{wt} = 1$ or the transfer car finishes its service if $N_{ws} > 1$. Thus, the service rate of FJQN can be calculated by:

$$\mu_f\left(N_f\right) = \sum_{N_{ws}=1}^{N_f} \frac{\pi(N_{ws}, 0)}{E(T_2)} + \sum_{N_{ws}=0}^{N_f-1} \frac{\pi(N_{ws}, 1)}{E(T_1)} \tag{12}$$

where $\pi(st_q)$ denotes the state probability of state st_q. Obviously, the state probabilities $\pi(st_q)$ can be obtained by solving the following:

$$\begin{cases} \boldsymbol{\pi Q} = 0 \\ \boldsymbol{\pi e} = 1 \end{cases} \tag{13}$$

where $\boldsymbol{e}$ is the column vector of ones, $\boldsymbol{Q}$ denotes the transition matrix of st_q (S1 File).

4.2 Solution to the closed queueing network

After obtaining the service rate of FJQN, we replace the FJQN by a FES node with exponential distributed load-dependent service time, $\mu_f(N_f)$. Then the network made up by all nodes (except for the synchronization node), are considered as a closed network (Fig 7b). Since class switching is allowed in this closed queueing network, the switch from classes to chains is needed [30]. According to the concept of chains, there are two chains in the closed queueing network denoted by c_1 and c_2. Without loss of generality, we let $c_1 = \{1\}$ represents storage transactions, $c_2 = \{2,3\}$ represents retrieval transactions and N_k denotes the number of customers in the closed network. The size of the state space therefore is reduced to:

$$\begin{pmatrix} 4 \cdot |c_1| + N_{k1} - 1 \\ 4 \cdot |c_1| - 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \cdot |c_2| + N_{k2} - 1 \\ 4 \cdot |c_2| - 1 \end{pmatrix}$$

where $|c_u|$ and N_{ku} denote the number of elements and customers in chain c_u, $u = 1,2$, respectively.

The routing probabilities are given by:

$$\begin{aligned} &p_{f1,31} = 1, p_{31,41} = 1, p_{41,f1} = 1 \\ &p_{f2,32} = 1, p_{32,52} = p_{sin}p_{ss}, p_{32,f3} = p_{sin}p_{sd} + p_{sio}, p_{52,f2} = 1 \\ &p_{f3,33} = 1, p_{33,53} = 1, p_{53,f2} = 1 \end{aligned}$$

The visit ratios in a chain are given by:

$$e_{ir} = \sum e_{js}p_{js,ir} for \begin{cases} r, s \in c_u \\ i, j = f, 3, 4, 5 \\ u = 1, 2 \end{cases} \tag{14}$$

After solving the system of linear Eq (14), we can get the visit ratios of the chain to and expected service time at node i:

$$e_{iu}^{c} = \frac{\sum_{r \in c_u} e_{ir}}{\sum_{r \in c_u} e_{fr}} \tag{15}$$

$$T_{iu}^{c} = \sum_{r \in c_u} T_{ir} \cdot \frac{e_{ir}}{\sum_{s \in c_u} e_{is}} \tag{16}$$

Then the throughputs of the closed queueing network, $\mu_1(N_k)$ for c_1 and $\mu_2(N_k)$ for c_2, are obtained through mean value analysis (MVA).

4.3 Steady state model of reduced SOQN

After substituting the subnetwork made up of all nodes with a FES node (Fig 7c), we first reduce the network into a single chain and then use a birth-death process to model the system [31]. Let the aggregate arrival rate $\lambda = \lambda_s + \lambda_r$ be the birth rate and the service rate for the aggregate chain $\mu(N_k) = p_s\mu_1(N_k) + p_r\mu_2(N_k)$ be the load-dependent death rate of the system. The state space is described using a single variable x, which represents the number of transactions waiting in queue Q_1 when $x > 0$ and the number of idle shuttles waiting in queue Q_2 when $-N_s \leq x \leq 0$. Thus, the load-dependent death rate $\mu(N_k) = \mu(N_s + x)$ when $-N_s \leq x \leq 0$ and $\mu(N_k) = \mu(N_s)$ when $x > 0$. The steady state probabilities can be obtained by using flow rate balance equations and can be expressed by (S2 File):

$$\pi(x) = \begin{cases} \pi(-N_s)\prod_{i=-N_s+1}^{x} \dfrac{\lambda}{\mu(N_s+i)}, & -N_s < x \leq 0 \\ \pi(-N_s)\prod_{i=-N_s+1}^{0} \dfrac{\lambda}{\mu(N_s+i)} \cdot \left(\dfrac{\lambda}{\mu(N_s)}\right)^{x}, & x > 0 \end{cases} \tag{17}$$

$$\pi(-N_s) = \frac{1}{1 + \sum_{i=1}^{N_s-1} \prod_{k=1}^{i} \frac{\lambda}{\mu(k)} + \left(\frac{\mu(N_s)}{\mu(N_s)-\lambda}\right) \prod_{k=1}^{N_s} \frac{\lambda}{\mu(k)}} \tag{18}$$

4.4 Performance measures of the system

The expected throughput time of the system, average utilizations of shuttles and transfer car, average queue length of Q_1 are the main performance measures we are interested in and are obtained using the following equations:

$$U_{sh} = \frac{N_s - \sum_{i=-N_s}^{-1} (-i)\pi(i)}{N_s} \tag{19}$$

$$U_t = \sum_{x_t \in X} p(x_t) \tag{20}$$

$$L_o = \pi(0) \cdot \frac{\lambda/\mu(N_s)}{[1 - \lambda/\mu(N_s)]^2} \tag{21}$$

$$E[T] = \frac{\sum_{k=1}^{\infty} k\pi(k) + N_s - \sum_{i=-N_s}^{-1} (-i)\pi(i)}{\lambda} \tag{22}$$

Table 5. Scenarios generated for model validation.

Depth / Width ratio	Total number of storage positions	Number of shuttles	Arrival rate of transactions	Number of scenarios
1:1	5000	2,3,4,5	22,25,28	12
1:1	10000	2,3,4,5	22,25,28	12
2:1	5000	2,3,4,5	22,25,28	12
2:1	10000	2,3,4,5	22,25,28	12

https://doi.org/10.1371/journal.pone.0259773.t005

Where $p(x_t)$ denotes the probability corresponding to the generic state x_t belonging to $\boldsymbol{X}$, the set of all possible states x of the system, x_t represents the states with the average number of transactions at FJQN, $L_f > 0$, and node 3, $L_3 > 0$.

5 Model validation and numerical experiments

5.1 Model validation

The simulation model is based on Arena software 14.0. Table 5 provides the details about the scenarios in simulation model. The data we use in the validation are derived from the study of Tappia et al. [2] and are provided in Table 6. The depth and width of a tier are measured by the maximum travel distance in the x- and y-direction, respectively. To validate the analytical model under different resource utilizations, the arrival rate of transactions is set at three levels: 22, 25 and 28 per hour with the assumption that $\lambda_r = \lambda_s$, which results a bottleneck utilization ranging from 70% to 90%. 12 scenarios are designed based on the variation of shuttle number and order arrival rate for each combination of depth / width ratio and total number of storage positions. Other assumptions are the same as the analytical model (i.e., POSC dwell point and random storage policy) (S3 File).

For each scenario, a warm-up period of more than 5000 transactions is run, followed by 15 replications with a run time of more than 30000 transactions, which leads to a 95% confidence interval where the half-width is less than 2% of the average. Four performance measures are estimated to validate the analytical model: the throughput time of system, the utilizations of transfer car and shuttles and the queue length of Q_1. The accuracy of analytical model is measured by absolute relative error, ε, which is defined as $\varepsilon = |A - S|/S \times 100\%$, where A and S denote the analytical and simulation results respectively. The computational complexity of the proposed model can be characterized by $O(N_s^4 \cdot \max(N_b, N_c)^2)$. In our experiments, the conduction of proposed algorithm takes less than 1 second of computational time on a standard computer.

The distribution of absolute relative errors for each performance measure is shown in Fig 8. The average absolute errors are 6.32%, 2.93%, 2.38% and 10.81% for expected throughput time, transfer car utilization, shuttle utilization and expected queue length of Q_1, respectively. These results suggest that the analytical model can accurately estimate the system performance.

Table 6. Data used for model validation.

Variable	Description	Value
w	Unit width per storage position	0.9m
d	Unit depth per storage position	1.2m
t_t, t_{sh}	Constant time required for transfer car or shuttle to load/unload the shuttle or unit load	5s
v_t, v_{sh}	Constant velocity of transfer car and shuttle	1m/s

https://doi.org/10.1371/journal.pone.0259773.t006

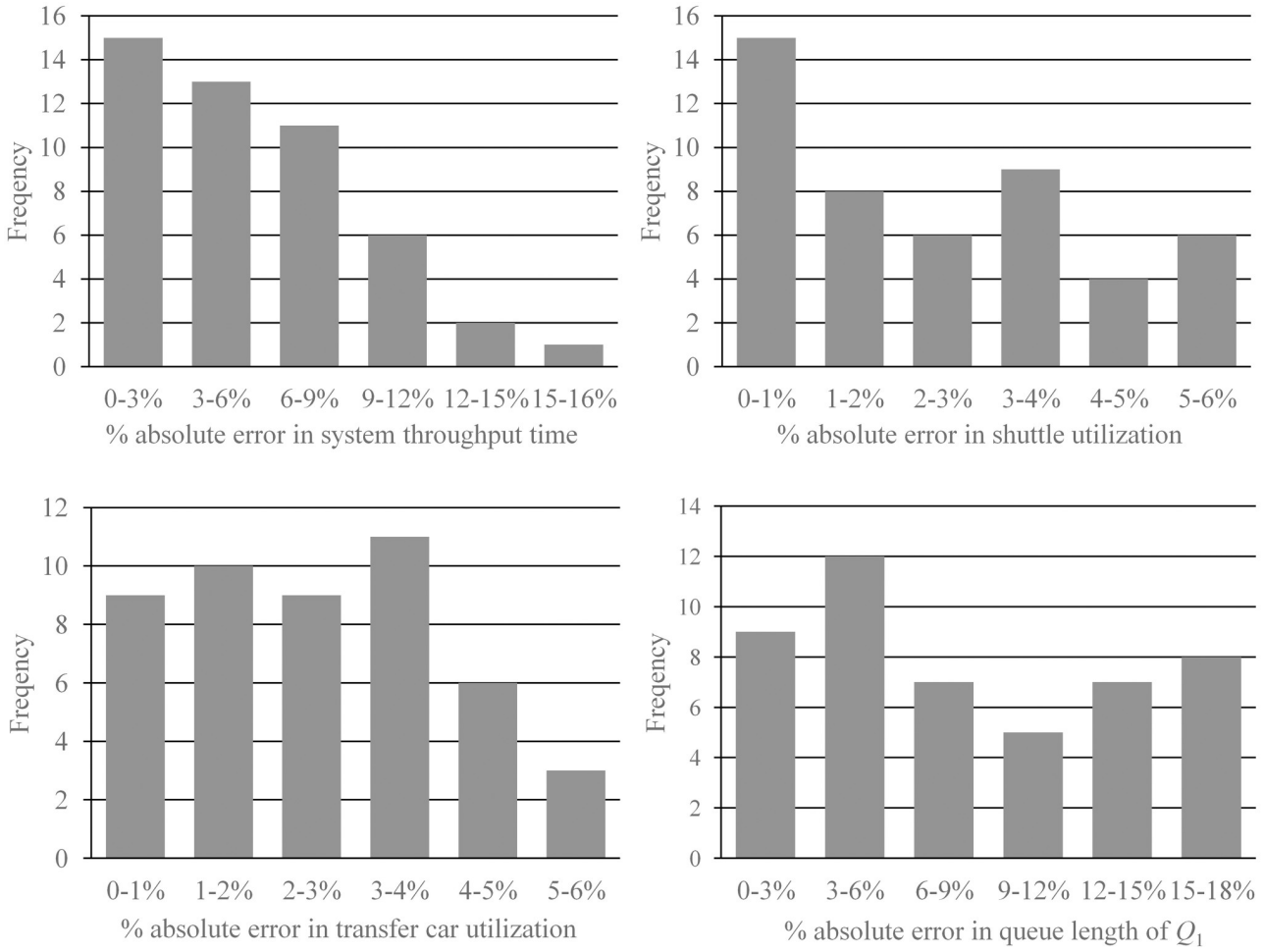

Fig 8. Distribution of absolute errors for performance measures.

https://doi.org/10.1371/journal.pone.0259773.g008

5.2 Comparison of sequential and parallel processing policies

As pointed out by Tappia et al. [2], it may be advantageous for deep lane shuttle-based compact storage systems under parallel processing policy, which means the response time of systems under parallel processing policy may be shorter than that of sequential processing policy with the increase of depth / width ratio. Thus, we compared the performance of two processing policies by carrying out numerical experiments. The system performance under sequential processing policy is estimated using simulation model, while the system performance under parallel processing policy is estimated using analytical model proposed in this study. To compare these two processing polices in more detail, we vary N_s and λ, i.e., N_s ranges from 2 to 5 and λ varies from18 to 30 with a step size of 0.1 to deal with the uncertainty of order arrival rate and we also assume that $\lambda_r = \lambda_s$. The total number of storage positions is 5000. The depth / width ratio varies from 0.75 to 3.5 with a step size of 0.25. The results are shown in Fig 9.

To better understand the difference in system throughput time under different processing policies, we use the average improvement percentage of the parallel processing policy over the

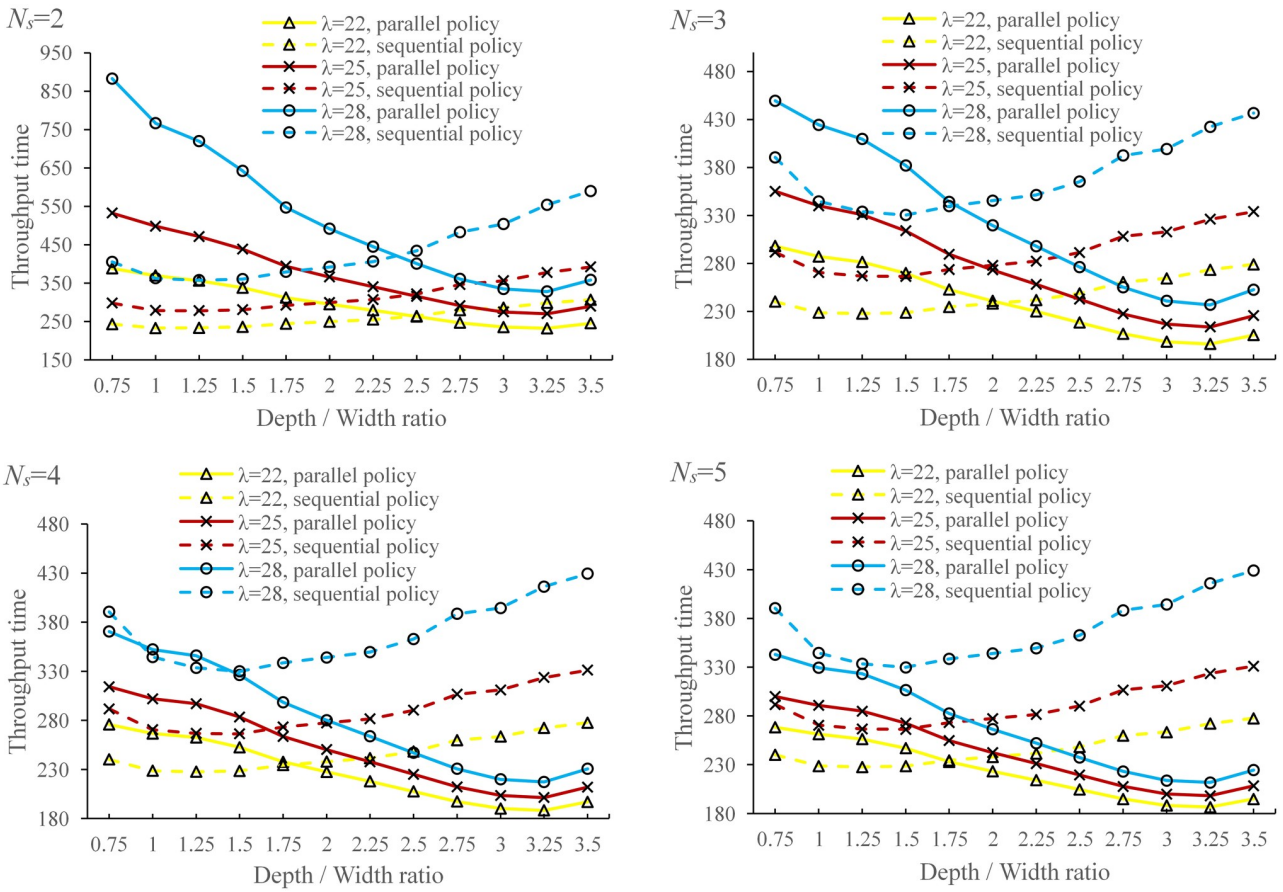

Fig 9. Comparison of parallel and sequential processing policies.

https://doi.org/10.1371/journal.pone.0259773.g009

sequential processing policy, I_P, which is defined by:

$$I_p = \frac{1}{V}\sum_{\lambda}\frac{E[T_S] - E[T_P]}{E[T_S]} \times 100\% \tag{23}$$

where $E[T_P]$ and $E[T_S]$ represent the expected system throughput time under parallel and sequential processing policy, respectively. And V is the number of values taken by λ. The results are shown in Fig 10.

Obviously, as shown in Fig 7, the average improvement percentage I_P increases with the depth / width ratio and the number of shuttles and the parallel policy performs better than sequential policy when depth / width ratio is large enough. Specifically, there is an intersection point between the curves of two processing policies, denoted by $(dw^*, E[T]^*)$. When $N_s = 2$, $\lambda = 28$, for example, $dw^* \approx 2.37$. When $dw > dw^*$, the parallel policy outperforms sequential policy. Additionally, dw^* decreases with the increase of N_s and λ.

Allowing the transfer car and shuttles to operate simultaneously reduces the total processing time, while its effect on total waiting time depends on the number of shuttles and depth / width ratio. Specifically, the processing time of parallel task is the maximum of shuttle processing time and transfer car processing time. At the meantime, for systems with storage lanes that are not too deep (i.e., $dw < dw^*$), the parallel policy increases the total waiting time due to a long travel distance of transfer car to pick up the waiting shuttles. This implies that the shuttles

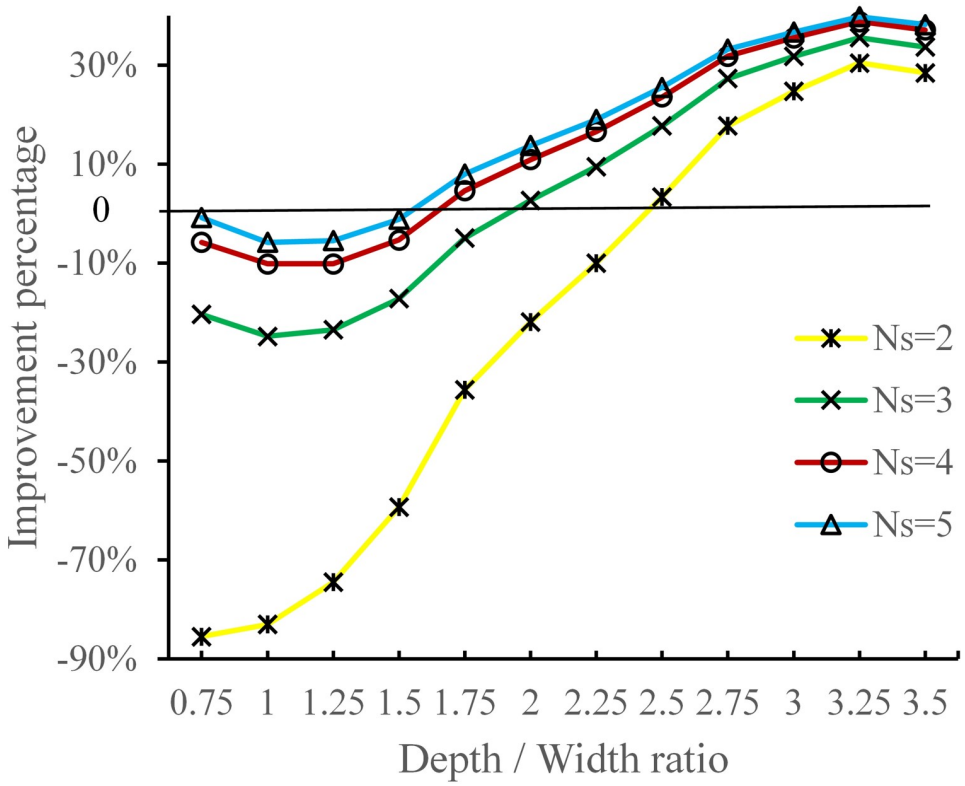

Fig 10. Average improvement percentage of the parallel processing policy over sequential processing policy.

https://doi.org/10.1371/journal.pone.0259773.g010

may always waiting for the service of transfer car, which resulting a higher utilization of transfer car and a longer waiting time of shuttles. For deep lane storage systems (i.e., $dw > dw^*$), the situation reverses since the capacity of transfer car is sufficient so that the increase of shuttle waiting time is dominated by the reduction of total processing time. Therefore, the performance of parallel policy is better than that of sequential policy. In addition, increasing the number of shuttles can reduce the total waiting time since the expected travel distance of transfer car is shorter. Thus, the reduction of total processing time can offset the increase of total waiting time easier (i.e., dw^* decreases).

5.3 Investigation of a real case

In this section, we estimate the performance of both sequential and parallel processing policies in a real case, which refers to a Nedcon system in UK [2]. The system consists of multiple tiers of multiple storage lanes with a layout as studied in our research. In each tier, there are 37 storage columns and 47 storage lanes at each side of the cross-aisle. As analyzing the real case, we should consider the effects of acceleration/deceleration of shuttles and the transfer car. Thus, the model has been adjusted to accommodate for acceleration/deceleration effects, which is referred to the work of Zou et al. [27]. And we also assume that $\lambda_r = \lambda_s$. Other system parameters are described in Table 7.

As shown in previous discussion, the depth/width ratio, transaction arrival rate and number of shuttles may affect the performance of sequential and parallel processing policies. Thus, for the analysis of a real case, we first vary the transaction arrival rate, ranging from 10 to 28 with a step size of 1, and keep the other variables fixed to investigate potential improvement in

Table 7. System parameters related to the real case.

Variable	Description	Value
w	Unit width per storage position	1.47m
d	Unit depth per storage position	0.9m
t_t, t_{sh}	Transfer car or shuttle loading/unloading time	3.5s; 6s
v_t, v_{sh}	Maximum velocity of transfer car and shuttle	1m/s
a_t, a_{sh}	Transfer car and shuttle acceleration/deceleration	0.3 m/s^2; 0.4 m/s^2

https://doi.org/10.1371/journal.pone.0259773.t007

throughput capacity as a result of adopting parallel processing policy with different resources utilizations. The results are showed in Fig 11.

With the increasing of transaction arrival rate, the increasing utilization of the transfer car increases the waiting time of shuttles for the service of the transfer car. Given the current configuration of the real system, the expected throughput capacity of sequential processing policy is larger than that of parallel processing when transaction arrival rate is relatively small. This may result from that when λ is small, the average waiting time of shuttles for the service of the transfer car is longer under parallel processing policy than that under sequential processing policy. For a large arrival rate, the situation reverses since the increase of shuttle waiting time, under parallel processing policy, is dominated by the reduction of total processing time. The

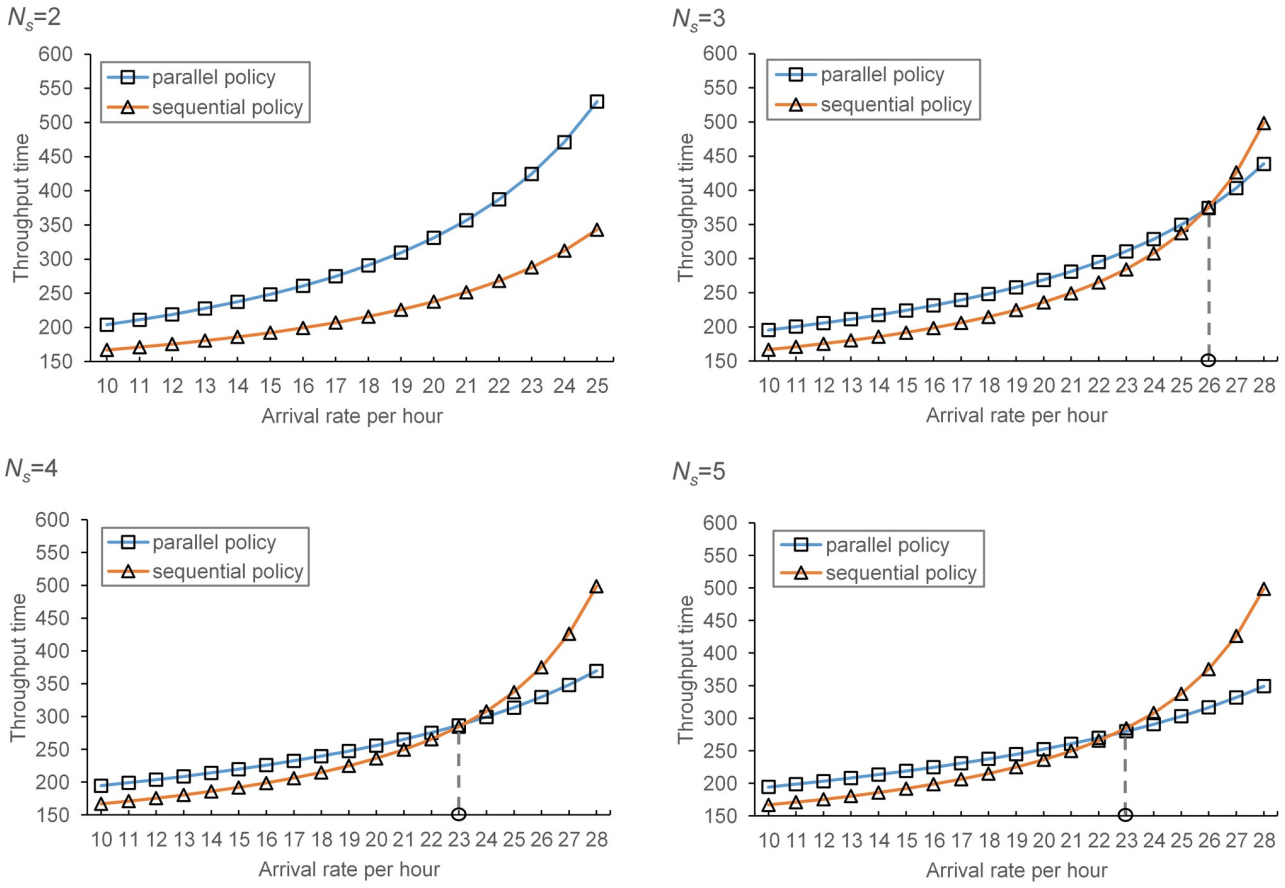

Fig 11. Comparison of parallel and sequential processing policies with different arrival rate.

https://doi.org/10.1371/journal.pone.0259773.g011

intersection point between the curves of the sequential processing policy and the parallel processing policy shows the critical transaction arrival rate, below which the sequential processing policy outperforms the parallel processing policy. On the other hand, when the number of shuttles increases, the critical transaction arrival rate decreases since adding new shuttles may shorten the average shuttle waiting time for the service of transfer car. Specifically, the critical transaction arrival rate is about 26 per hour when $N_s = 3$, and about 23 when $N_s = 4, 5$. For the case of two shuttles, the critical transaction arrival rate is larger than 28 per hour, where the resource utilizations are higher than 95% and may not guarantee the conditions for convergence of the system. Thus, we eliminate the scenarios with transaction arrival rates larger than 26.

For the analysis of the effects of tier configuration, we vary the depth/width ratio from 0.5 to 3.5 with a step size of 0.25, and keep the other variables fixed (the transaction arrival rate is 22 per hour). The results are provided in Fig 12.

The optimal depth/width ratio under parallel processing policy is 1.75, larger than that of sequential processing policy. This implies that, given the current system configurations, the maximum system throughput can be achieved when the depth/width ratio is 1.75 and the system throughput decreases as the depth/width ratio increase or decrease. And the curve is very flat at the optimal ratio point. As discussed in previous section, there exists a critical depth/width ratio, below which the sequential processing policy outperforms the parallel processing policy. And adding new shuttles also results a decreasing of the critical depth/width ratio.

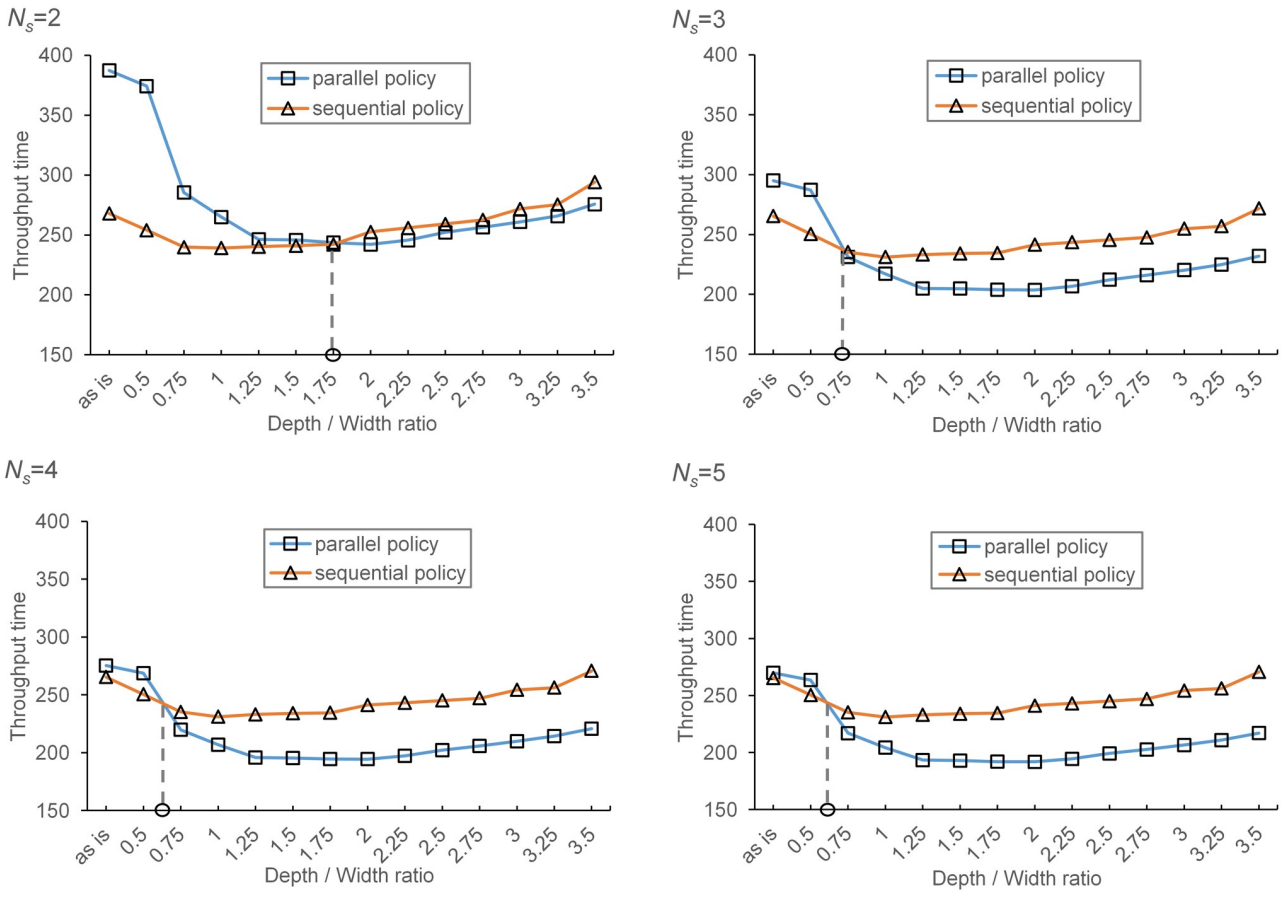

Fig 12. Comparison of parallel and sequential processing policies with different depth/width ratio.

https://doi.org/10.1371/journal.pone.0259773.g012

Specifically, the critical depth/width ratio is about 1.75 when $N_s = 2$, about 0.72 when $N_s = 3$, and about 0.625 when $N_s = 4$, 5.

Given the current configuration of the real system, the sequential processing policy outperforms the parallel processing policy. However, when the arrival rate of transactions becomes large (e.g., during COVID-19), the parallel processing policy should be considered. Our results also allow showing that the depth/width ratio have a significant impact on the difference in system performance between sequential and parallel processing policy. This implies the adoption of parallel processing policy may shorten system response time in the systems with deep storage lanes. Besides, despite the increase of investment cost, adding new shuttles may be a useful way to improve system performance since it will reduce the critical transaction arrival rate and encourage the transform of processing policy from sequential to parallel, which may further improve the system performance. For the system design, our results suggest that the optimal depth/width ratio should be used as a guiding factor.

To better understand the system performance under different processing policies, we set the depth/width ratio at the optimal level (i.e., 1.75 for system under parallel processing policy and 1.25 for sequential processing policy), vary the transaction arrival rate, ranging from 10 to 28 with a step size of 1 and keep the other variables fixed. The results are provided in Fig 13.

When the number of shuttles is small (i.e., $N_s = 2$) and the arrival rate of transactions is relatively low (smaller than 24 per hour), the system throughput under sequential processing policy is better than that under parallel processing given the current system configurations and

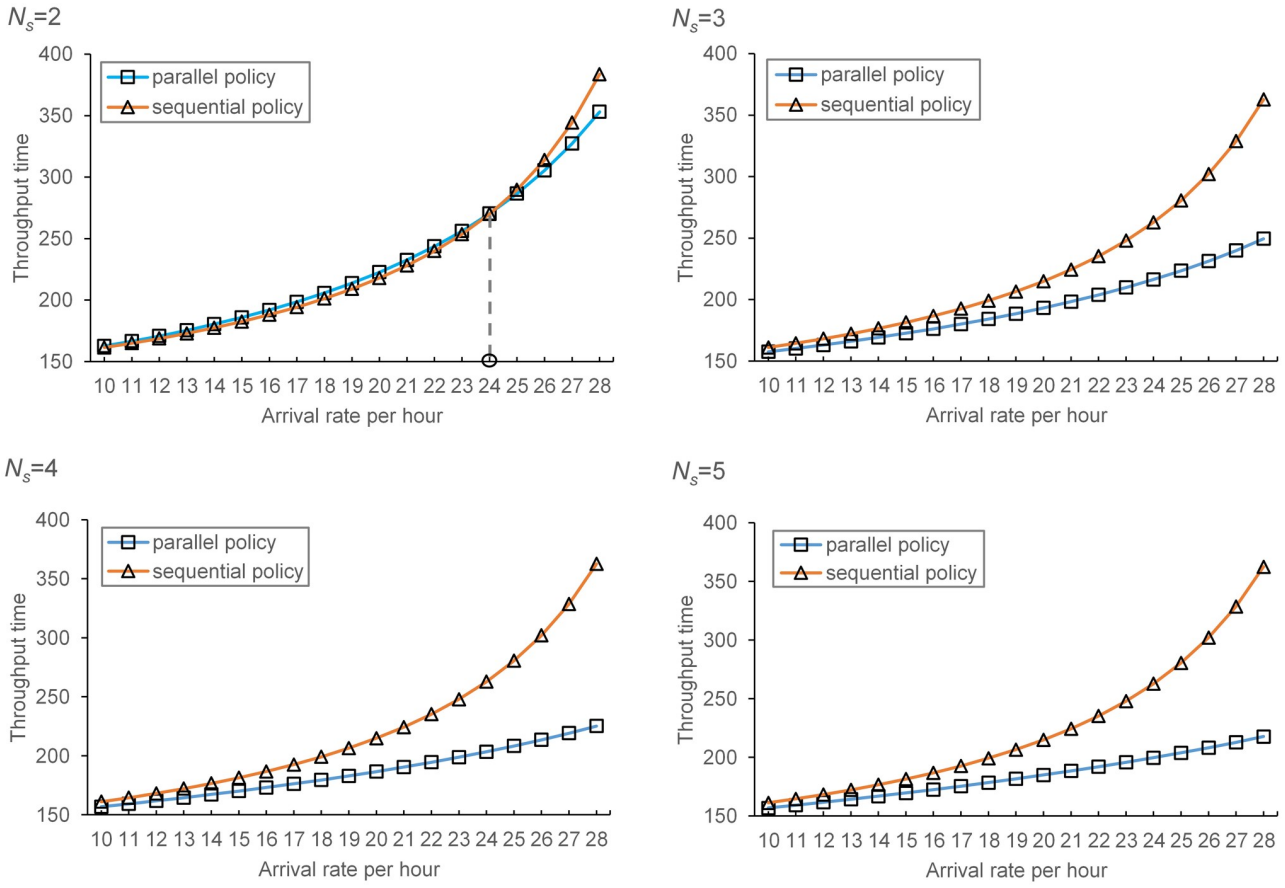

Fig 13. Comparison of parallel and sequential processing policies under the optimal depth/width ratio.

https://doi.org/10.1371/journal.pone.0259773.g013

the optimal depth/width ratio (i.e., 1.75 for system under parallel processing policy and 1.25 for sequential processing policy). However, when adding new shuttles or the arrival rate of transactions becomes larger, the parallel policy outperforms the sequential processing policy. Besides, the advantage of the parallel processing policy increases with the increase of shuttle number and the transaction arrival rate. These results suggest that, considering the variety of customer demands, the parallel processing policy should be considered and the optimal depth/width ratio of parallel processing policy should be used as a guiding factor.

6 Conclusions and future works

The shuttle-based compact storage systems are becoming popular and adopted by many modern warehouses. Considering the variety of customer demands, it is important to improve the performance of such systems. Given its advantages in improving system performance, the parallel processing policy in shuttle-based compact storage systems need to be investigated. However, studies on this subject are rare. This study is one of the first to estimate the system performance of parallel processing policy in shuttle-based compact systems. Our contributions lie in both developing an analytical model and providing operational and design insights. Specifically, we mainly focus on the performance estimation of a single-tier of specialized shuttle-based compact storage system, in which the shuttles can only move within storage lanes and are transported along the cross-aisle by the transfer car, under parallel processing policy. The system is modelled as a multi-class semi-open queuing network with class switching, so that transfer car can perform other tasks during a retrieval transaction. Both storage and retrieval transactions are considered to capture the dynamic of shuttle routes and estimate the effect of different transactions on system throughput time. To capture the effect of simultaneously operations of the shuttles and the transfer car, we formulate a FJQN in which the transaction will be split into two parts, one is served by the shuttle and the other is served by the transfer car. Since exact solutions to the proposed semi-open queuing network are not available, a decomposition-based approach is developed to estimate the performance of the system. The analytical model is validated against simulations, the average errors for system response time, shuttle and transfer car utilization and external queue length are 6.32%, 2.93%, 2.38% and 10.81%, respectively.

We carry out a series of numerical experiments to compare the performance of sequential and parallel processing policies. The results show that the parallel processing policy outperforms the sequential processing policy in systems with deep storage lanes (which means the depth/width ratio of the system is large). Additionally, the advantage of the parallel processing policy increases with the increase of shuttle number, the depth/width ratio and the transaction arrival rate. Our results also show that there is a critical depth/width ratio, below which the system should follow the sequential processing policy. Otherwise, the parallel processing policy should be considered. We also investigate the performance of both sequential and parallel processing policies in a real case. Given the current configuration of the real system, the system response time of sequential processing policy is lower than that of parallel processing policy. However, when the transaction arrival rate becomes large, our results suggest benefits of adopting parallel processing policy. The results also show the critical point of transaction arrival rate and depth/width ratio under different shuttle numbers. Besides, the optimal depth/width ratio of the real system is 1.75 when parallel processing policy is used, which is independent of the shuttle number and the transaction arrival rate. When comparing the system performance of different processing policy under the optimal depth/width ratio (1.75 for parallel and 1.25 for sequential processing policy), the results show that the sequential processing policy only have advantages when there are two shuttles in a tier and the transaction arrival rate is

small (smaller than 24 per hour). This suggests a potential improvement in system performance achieved by the adoption of parallel processing policy considering the variety of customer demand.

This study provides some useful managerial implications and warehouse design insights. However, there is nevertheless a set of limitations. First, the proposed model is only applied on only one real system. Thus, the findings, such as the optimal and critical depth/width ratio and the potential improvement in system performance achieved by adopting parallel processing policy, may not be applicable to other warehouses with different system configurations. Second, in order to develop a tractable model, some assumptions are made in this study, such as random storage policy, POSC dwell point policy, FCFS scheduling policy and so on, all of which could be relaxed. therefore, for future research, it is interesting to consider the effect of different storage assignment policies, different dwell point policies, different transaction scheduling policies, different shuttle assignment rules and the blocking effects. Additionally, it would be interesting to investigate the system performance with transactions requiring more than one unit load and considering both single- and dual-command cycles. On the other hand, future research would include applying the proposed model on other systems where resources work simultaneously and developing more accurate and robust modeling approaches.

Supporting information

S1 File. Details on the component of transition matrix *Q* of FJQN.
(PDF)

S2 File. Details on the solution approach for reduced network with a single server.
(PDF)

S3 File. Details of simulation models.
(PDF)

Acknowledgments

We acknowledge the study of Tappia and his colleagues in supporting our research with real data.

Author Contributions

Conceptualization: Lei Deng, Ruimei Wang.

Formal analysis: Lei Deng, Jingjie Zhao.

Funding acquisition: Lei Deng.

Methodology: Lei Deng, Jingjie Zhao, Ruimei Wang.

Software: Lei Deng, Lei Chen.

Supervision: Lei Deng, Ruimei Wang.

Validation: Jingjie Zhao.

Visualization: Lei Chen, Jingjie Zhao.

Writing – original draft: Lei Deng.

Writing – review & editing: Lei Deng, Lei Chen.

References

1. Azadeh K, De Koster R, Roya D. Robotized and Automated Warehouse Systems: Review and Recent Developments. Transportation Science. 2019; 53(4): 917–945. https://doi.org/10.1287/trsc.2018.0873
2. Tappia E, Roy D, Koster R, Melacini M. Modeling, Analysis, and Design Insights for Shuttle-Based Compact Storage Systems. Transportation Science. 2017; 51(1): 269–295. https://doi.org/10.1287/trsc.2016.0699
3. Borovinšek M, Ekren BY, Burinskienė A, Lerher T. Multi-Objective Optimisation Model of Shuttle-Based Storage and Retrieval System. Transport. 2017; 32(2): 120–137. https://doi.org/10.3846/16484142.2016.1186732
4. Ekren BY. Graph-Based Solution for Performance Evaluation of Shuttle-Based Storage and Retrieval System. International Journal of Production Research. 2017; 55(21): 6516–6526. https://doi.org/10.1080/00207543.2016.1203076
5. Malmborg CJ. Conceptualizing Tools for Autonomous Vehicle Storage and Retrieval Systems. International Journal of Production Research. 2002; 40(8): 1807–1822. https://doi.org/10.1080/00207540110118668
6. Malmborg CJ. Design Optimization Models for Storage and Retrieval Systems Using Rail Guided Vehicles. Applied Mathematical Modelling. 2003; 27(12): 929–941. https://doi.org/10.1016/S0307-904X(03)00127-6
7. Malmborg CJ. Interleaving Dynamics in Autonomous Vehicle Storage and Retrieval Systems. International Journal of Production Research. 2003; 41(5): 1057–1069. https://doi.org/10.1080/0020754021000033887
8. Fukunari M, Malmborg CJ. An Efficient Cycle Time Model for Autonomous Vehicle Storage and Retrieval Systems. International Journal of Production Research. 2008; 46(12): 3167–3184. https://doi.org/10.1080/00207540601118454
9. Kuo P, Krishnamurthy A, Malmborg CJ. Design Models for Unit Load Storage and Retrieval Systems Using Autonomous Vehicle Technology and Resource Conserving Storage and Dwell Point Policies. Applied Mathematical Modelling. 2007; 31(10): 2332–2346. https://doi.org/10.1016/j.apm.2006.09.011
10. Roy D, Krishnamurthy A, Heragu SS, Malmborg CJ. Performance Analysis and Design Trade-Offs in Warehouses with Autonomous Vehicle Technology. IIE Transactions. 2012; 44(12): 1045–1060. https://doi.org/10.1080/0740817x.2012.665201
11. Heragu SS, Cai X, Krishnamurthy A, Malmborg CJ. Analytical Models for Analysis of Automated Warehouse Material Handling Systems. International Journal of Production Research. 2011; 49(22): 6833–6861. https://doi.org/10.1080/00207543.2010.518994
12. Marchet G, Melacini M, Perotti S, Tappia E. Analytical Model to Estimate Performances of Autonomous Vehicle Storage and Retrieval Systems for Product Totes. International Journal of Production Research. 2012; 50(24): 7134–7148. https://doi.org/10.1080/00207543.2011.639815
13. Ekren BY, Akpunar A. An Open Queuing Network Based Tool for Performance Estimations in a Shuttle-Based Storage and Retrieval System. Applied Mathematical Modelling. 2021; 89: 1678–1695. https://doi.org/10.1016/j.apm.2020.07.055
14. Zhao X, Zhang R, Zhang N, Wang Y, Jin M, Mou S. Analysis of the Shuttle-Based Storage and Retrieval System. IEEE Access. 2020; 8: 146154–146165. https://doi.org/10.1109/ACCESS.2020.3014102
15. Ha Y, Chae J. Free Balancing for a Shuttle-Based Storage and Retrieval System. Simulation Modelling Practice and Theory. 2018; 82: 12–31. https://doi.org/10.1016/j.simpat.2017.12.006
16. Wu Y, Zhou C, Ma W, Kong XTR. Modelling and Design for a Shuttle-Based Storage and Retrieval System. International journal of production research. 2020; 58(16): 4808–4828. https://doi.org/10.1080/00207543.2019.1665202
17. Lei B, Hu F, Jiang Z, Mu H. Optimization of Storage Location Assignment in Tier-To-Tier Shuttle-Based Storage and Retrieval Systems Based on Mixed Storage. Mathematical Problems in Engineering. 2020: 1–17. https://doi.org/10.1155/2020/2404515
18. Luo L, Zhao N, Lodewijks G. Scheduling Storage Process of Shuttle-Based Storage and Retrieval Systems Based on Reinforcement Learning. Complex System Modeling and Simulation. 2021; 1(2): 131–144. https://doi.org/10.23919/CSMS.2021.0013
19. Dong W, Jin M, Wang Y, Kelle P. Retrieval Scheduling in Crane-Based 3D Automated Retrieval and Storage Systems with Shuttles. Annals of Operations Research. 2021; 302(1): 111–135. https://doi.org/10.1007/s10479-021-03967-8
20. Liu Z, Wang Y, Jin M, Wu H, Dong W. Energy Consumption Model for Shuttle-Based Storage and Retrieval Systems. Journal of Cleaner Production. 2021; 282: 124480. https://doi.org/10.1016/j.jclepro.2020.124480

21. 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). International Journal of Production Research. 2016; 54(14): 4286–4304. https://doi.org/10.1080/00207543.2016.1144241

22. D'Antonio G, Maddis MD, Bedolla JS, Chiabert P, Lombardi F. Analytical Models for the Evaluation of Deep-Lane Autonomous Vehicle Storage and Retrieval System Performance. The International Journal of Advanced Manufacturing Technology. 2018; 94(5–8): 1811–1824. https://doi.org/10.1007/s00170-017-0313-2

23. Boysen N, Boywitz D, Weidinger F. Deep-Lane Storage of Time-Critical Items: One-Sided Versus Two-Sided Access. OR Spectrum. 2018; 40(4): 1141–1170. https://doi.org/10.1007/s00291-017-0488-9

24. Eder M. An Approach for a Performance Calculation of Shuttle-Based Storage and Retrieval Systems with Multiple-Deep Storage. The International Journal of Advanced Manufacturing Technology. 2020; 107(1–2): 859–873. https://doi.org/10.1007/s00170-019-04831-7

25. Kumawat G. L., Roy D., 2021. A New Solution Approach for Multi-Stage Semi-Open Queuing Networks: An Application in Shuttle-Based Compact Storage Systems. Computers & Operations Research. 125: 105086. https://doi.org/10.1016/j.cor.2020.105086

26. Hu Y, Huang SY, Chen C, Hsu W, Toh AC, Loh CK, et al. Travel Time Analysis of a New Automated Storage and Retrieval System. Computers & Operations Research. 2005; 32(6): 1515–1544. https://doi.org/10.1016/j.cor.2003.11.020

27. Zou B, Xu X, Yale Gong Y, De Koster R. Modeling Parallel Movement of Lifts and Vehicles in Tier-Captive Vehicle-Based Warehousing Systems. European Journal of Operational Research. 2016; 254(1): 51–67. https://doi.org/10.1016/j.ejor.2016.03.039

28. Kumawat GL, Roy D, De Koster R, Adan I. Stochastic Modeling of Parallel Process Flows in Intra-logistics Systems: Applications in Container Terminals and Compact Storage Systems. European Journal of Operational Research. 2021; 290(1): 159–176. https://doi.org/10.1016/j.ejor.2020.08.006

29. Jia J, Heragu SS. Solving Semi-Open Queuing Networks. Operations Research. 2009; 57(2): 391–401. https://doi.org/10.1287/opre.1080.0627

30. Bolch G, Greiner S, de Meer H, Trivedi KS. Queueing Networks and Markov Chains. Wiley-Interscience, New Jersey; 2006.

31. Roy D, Bandyopadhyay A, Banerjee P. A Nested Semi-Open Queuing Network Model for Analyzing Dine-In Restaurant Performance. Computers & Operations Research. 2016; 65: 29–41. https://doi.org/10.1016/j.cor.2015.06.006