Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems

Bipan Zou, Xianhao Xu, Yeming Gong, René de Koster

▶ **To cite this version:**

Bipan Zou, Xianhao Xu, Yeming Gong, René de Koster. Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems. European Journal of Operational Research, 2016. hal-01892897

HAL Id: hal-01892897
https://hal.science/hal-01892897

Submitted on 10 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Accepted Manuscript

Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems

Bipan Zou, Xianhao Xu, Yeming (Yale) Gong, René De Koster

PII: S0377-2217(16)30167-9
DOI: 10.1016/j.ejor.2016.03.039
Reference: EOR 13601

To appear in: *European Journal of Operational Research*

Received date: 6 August 2015
Revised date: 18 March 2016
Accepted date: 21 March 2016

Please cite this article as: Bipan Zou, Xianhao Xu, Yeming (Yale) Gong, René De Koster, Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems, *European Journal of Operational Research* (2016), doi: 10.1016/j.ejor.2016.03.039

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Highlights

- We propose a parallel policy for autonomous tier-captive storage/retrieval systems.
- A fork-join queueing network is formulated to analyze the system performance.
- The parallel policy has advantages in small size systems.
- The parallel policy has advantages below a critical transaction arrival rate.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems

Bipan Zou[a], Xianhao Xu[a,*], Yeming (Yale) Gong[b], René De Koster[c]

[a] *School of Management, Huazhong University of Science and Technology, Wuhan, 430074, China*
[b] *EMLYON Business School, 23 avenue Guy de Collongue, 69134 Ecully Cedex, France*
[c] *Rotterdam School of Management, Erasmus University, P.O.Box 1738, 3000DR Rotterdam, The Netherlands*

Abstract

This paper models and analyzes tier-captive autonomous vehicle storage and retrieval systems. While previous models assume sequential commissioning of the lift and vehicles, we propose a parallel processing policy for the system, under which an arrival transaction can request the lift and the vehicle simultaneously. To investigate the performance of this policy, we formulate a fork-join queueing network in which an arrival transaction will be split into a horizontal movement task served by the vehicle and a vertical movement task served by the lift. We develop an approximation method based on decomposition of the fork-join queueing network to estimate the system performance. We build simulation models to validate the effectiveness of analytical models. The results show that the fork-join queueing network is accurate in estimating the system performance under the parallel processing policy. Numerical experiments and a real case are carried out to compare the system response time of retrieval transactions under parallel and sequential processing policies. The results show that, in systems with less than 10 tiers, the parallel processing policy outperforms the sequential processing policy by at least 5.51%. The advantage of parallel processing policy is decreasing with the rack height and the aisle length. In systems with more than 10 tiers and a length to height ratio larger than 7, we can find a critical retrieval transaction arrival rate, below which the parallel processing policy outperforms the sequential processing policy.

Keywords: Logistics, warehousing, AVS/RS, analytical and simulation modelling, performance analysis

*Corresponding author
Email addresses: D201377812@hust.edu.cn (Bipan Zou), xxhao@mail.hust.edu.cn (Xianhao Xu), GONG@em-lyon.com (Yeming (Yale) Gong), rkoster@rsm.nl (René De Koster)

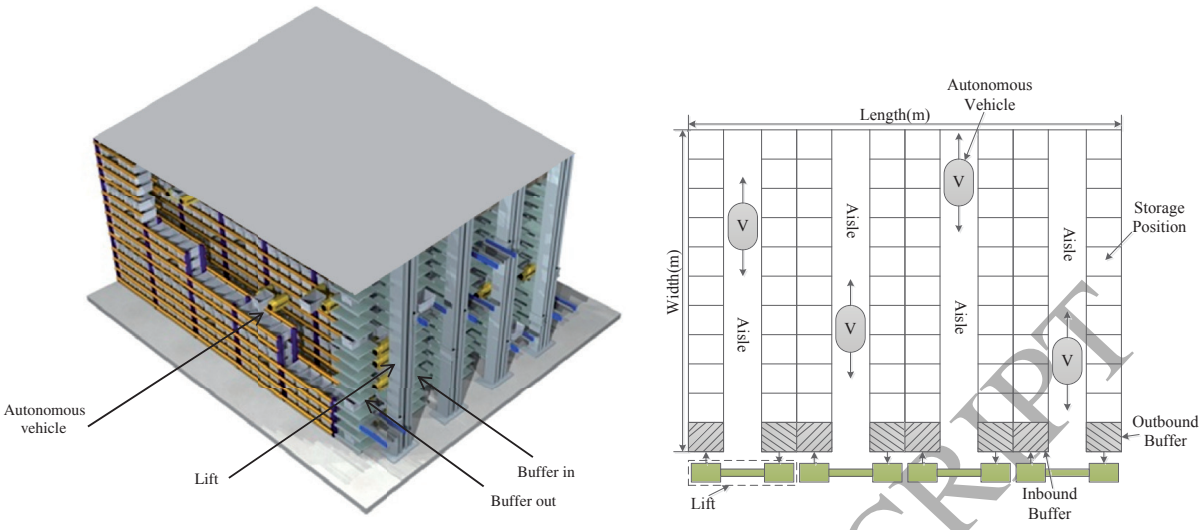
ACCEPTED MANUSCRIPT

1. Introduction

Autonomous vehicle-based storage and retrieval systems (AVS/RSs) can store and retrieve goods stored on unit loads with high operational efficiency and flexibility. These systems have been adopted by many distribution centers in their high density storage areas, since the introduction by Savoye Logistics at the beginning of the 20th century (Cai et al. (2014), Roy et al. (2015)).

Depending on the assignment of the vehicles to the tiers, AVS/RSs can be classified as tier-to-tier AVS/RSs where the vehicle can visit all tiers, and tier-captive AVS/RSs where the vehicle is captive to its designated tier (Heragu et al. (2011)). The tier-to-tier configuration is more suitable for handling palletized unit-loads with expensive vehicles (see Savoye Logistics). In such a system, the movement of loads should be matched with a vehicle until the completion of the transaction (Ekren et al. (2010)). The tier-captive configuration is more suitable for handling small-size unit loads, such as totes, with inexpensive vehicles (see KNAPP (2016)). In such a system, each vehicle can only move in its designated aisle (Marchet et al. (2012)). Fig.1 presents a typical tier-captive AVS/RS for product totes. It consists of three components: a rack system containing single-deep storage racks, lifts that are mounted along the periphery of the racks and autonomous vehicles that move in aisles. The input/output point (I/O) of the system is located at the first tier of each aisle. The first storage positions at each side of the aisle are buffer locations. One is the inbound buffer which holds the loads to be stored, and the other is the outbound buffer which holds the loads that have been retrieved. In the tier-captive configuration, each vehicle is dedicated to an aisle and it provides horizontal load movement. The lift is discrete and it holds one load once time. It provides not only the vertical load movement for the storage and retrieval transactions, but also the load transfer to and from the buffer at each tier.

Previous researches of tier-captive AVS/RSs follow a sequential processing policy (Heragu et al. (2011), Marchet et al. (2012), Marchet et al. (2013), Lerher (2015), Lerher et al. (2015a) and Lerher et al. (2015b)). In this policy, a storage transaction first requests the lift and then, sequentially, the vehicle, while a retrieval transaction requests the lift and vehicle in a reverse order. For a retrieval transaction, the vehicle in the designated tier transports the retrieval load to the outbound buffer and then the transaction requests the lift. For a storage transaction, the



(a) Isometric view(Marchet et al. (2012)) (b) Top view

Figure 1: Tier-captive AVS/RSs for product totes

lift transports the load to be stored from the I/O point to the inbound buffer of the designated tier, and then the transaction requests the vehicle. However, since the vehicle in a tier-captive configuration can only move in its designated tier, a parallel processing policy under which the transactions request the lift and the vehicle simultaneously, may perform better. Under such a policy, for a retrieval transaction, the vehicle transports the load to be retrieved to the outbound buffer and meanwhile, the lift moves to the designated tier. For a storage transaction, the lift transports the load to be stored to the inbound buffer of the designated tier. Meanwhile, the vehicle moves to the inbound buffer. Hu et al. (2005) have examined such a policy for a special type of automated storage/retrieval system-Split-Platform AS/RS (SP-AS/RS), in which the S/R machine consists of one vertical platform and N horizontal platforms. These systems differ from AVS/RSs, since the S/R machine can travel inside the aisles and take care of the vertical transport as well. In addition, it does not have a buffer area where the storage and retrieval loads can be hold temporarily. While Hu et al. (2005) use deterministic models, we formulate stochastic models to estimate the performance of tier-captive AVS/RSs, thereby taking into account the effect of waiting for the different resources.

To analyze the system performance under the parallel processing policy, we formulate a fork-join queueing network (FJQN) where a transaction will be split into a horizontal movement task served by the vehicle and a vertical movement task served by the lift. Based on the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

decomposition method of fork-join queueing networks presented in Bolch (2006), we develop an approximation method to estimate the system performance. We validate the effectiveness of our fork-join queueing network using simulation. The results show that our analytical model is accurate for performance estimation. The results of the numerical experiments show that the system performance under the parallel processing policy is sensitive to the rack structure and the transaction arrival rate.

The remainder of this paper is organized as follows: Section 2 presents related literature. Section 3 describes the system. In Section 4, we model the FJQN and introduce the methods applied for performance estimation. Section 5 contains the simulation validation and numerical experiments. Section 6 presents conclusions and future works.

2. Literature Review

In this section, we first present literature on models of tier-to-tier configurations, followed by literature on tier-captive AVS/RSs models. Finally, we discuss the parallel processing policy.

AVS/RSs with tier-to-tier configurations are widely used and studied. Malmborg (2002) is the first one to study tier-to-tier AVS/RSs, using continuous markov chain models for both horizontal and vertical material flows to calculate the expected S/R cycle time (weighted sum of single and dual-command cycle times). Malmborg (2003a) takes cost factors into consideration and formulates an analytical model aiming at developing useful system designs. Malmborg (2003b) proposes a state equation model to predict the proportion of dual-command cycles for systems with opportunistic-interleaving, i.e., combining storage and retrieval transactions on S/R cycles on an opportunistic basis.

Due to the impact of queuing in the system on performance, many papers have emerged using queuing models. Kuo et al. (2007) examine the random storage policy and the point-of-service-completion (POSC) dwell point policy in tier-to-tier AVS/RS. They model the vehicle service process as an $M/G/V$ queue and the lift service process as a $G/G/L$ queue. Fukunari & Malmborg (2008) consider both single and dual-command cycles and model the lift service process as an $M/G/L$ queue nested within an $M/G/V$ queue modeling the vehicle service process. They derive the transaction cycle time by iteratively calculating the percentage of dual-command cycles until convergence of the waiting time in these two queuing systems. Fukunari & Malmborg (2009) extend this work and formulate a closed queueing network for

ACCEPTED MANUSCRIPT

tier-to-tier AVS/RSs, including the maintenance and repair of vehicles. Zhang et al. (2009) propose an approximation method based on the variance of the transaction inter-arrival times, which can accurately estimate the transaction waiting time.

Since modeling the vehicles as an additional resource may be a better approach to capture the effect of the number of vehicles on system performance, recent studies model tier-to-tier systems as semi-open queueing networks (SOQN). Roy et al. (2012) build a multi-class SOQN with class switching for a single-tier AVS/RS, and design a decomposition method to estimate the system performance. Ekren et al. (2014) develop a matrix-geometric method to analyze the SOQN model built for tier-to-tier AVS/RSs. Cai et al. (2014) model a tier-to-tier system as a multi-class multi-stage SOQN, and use matrix-geometric methods to analyze it. Roy et al. (2015) investigate the position of the load/unload, the POSC dwell point policies and the location of cross-aisles in a single-tier AVS/RS. They also model the system as a SOQN. Tappia et al. (2016) study shuttle-based compact storage systems (AVS/RS with multi-deep storage lanes), considering both specialized and generic shuttles and both discrete and continuous lifts. They build multi-class SOQNs for both single and multi-tier systems and use matrix-geometric methods to analyze the SOQNs.

Different from tier-to-tier configurations, tier-captive configurations have drawn little attention. Heragu et al. (2011) formulate an open queueing network to estimate the performance of tier-captive AVS/RS and investigate the advantages of AVS/RS over traditional AS/RS. Also, Marchet et al. (2012) build an open queueing network to estimate the response time of a tier-captive AVS/RS used for product totes, considering acceleration and deceleration of lifts and vehicles. Marchet et al. (2013) design a comprehensive design framework for AVS/RSs. Application of the framework shows that tier-to-tier systems prefer a large number of short aisles and tier-captive systems prefer a small number of long aisles. Lerher et al. (2015a) study shuttle-based storage and retrieval systems (one lift tier-captive AVS/RS). They consider acceleration and deceleration of lifts and vehicles, and derive closed-form expression of both horizontal and vertical loads movement. Subsequently, Lerher (2015) extends this work to shuttle-based storage and retrieval systems with double-deep storage racks.

Our literature review shows that, previous studies of tier-captive configurations follow a sequential processing policy, under which the transfer time involving a vehicle and a lift, is the sum of the vertical movement time, the horizontal movement time, and the waiting time.

ACCEPTED MANUSCRIPT

However, since the vehicle and the lift can work as parallel servers (Marchet et al. (2012)), the transfer time will be the sum of the maximum of these two movement times and the waiting time. The impact of the parallel processing policy may be substantial, which justifies explicit inclusion in the models.

3. System Description and Modeling Preparations

We present the operational process of the tier-captive AVS/RSs under the parallel processing policy in Section 3.1. Section 3.2 summarizes the notations and the assumptions used in this paper. Section 3.3 describes the components of the transaction cycle time related to the lift and the vehicle.

In a tier-captive AVS/RS (see Fig.1), a discrete lift (consisting of one lifting table) transports unit loads between the tiers. Each aisle has a tier-captive vehicle (a dedicated shuttle) that transports loads into and out of individual storage positions and inbound/outbound buffers. The lift and the vehicles can move in parallel, as the loads are transferred via buffers (one for inbound loads and one for outbound loads in each aisle at each tier). Based on the system structure, the lift and the parts served by the lift can be considered as one unit of the system. This paper investigates the efficiency and advantages of parallel processing policy in tier-captive AVS/RSs. To this end, we examine one unit of the system (one lift system), which is equivalent to analyze the whole system (multiple lifts system). The storage and retrieval operations of the tier-captive AVS/RSs under the parallel processing policy can be described as follows. The flowcharts of the operational processes can be found in Appendix **A**.

For a storage transaction:

(1) The lift moves from its dwell point to the ground-floor tier, i.e., the I/O point of the system. Then, the lift picks up the storage load and goes to the designated tier. When the lift reaches its destination, it releases the load in the inbound buffer. Meanwhile, the vehicle in the designated tier moves from its dwell point to the inbound buffer.

(2) If the vehicle reaches the inbound buffer and the storage load is in the inbound buffer, the vehicle picks up the load. If either the vehicle or the storage load reaches the inbound buffer first, it has to wait for the other. After the vehicle has picked up the load, the vehicle brings it to the designated storage position and releases it.

ACCEPTED MANUSCRIPT

If the storage position is at the ground-floor tier, the lift should only move to the ground-floor tier, picks up the load at the I/O point and then releases the load in the inbound buffer of the ground-floor tier.

For a retrieval transaction:

(1) The vehicle in the designated tier moves from its dwell point to the retrieval load, picks up the load, and goes to the outbound buffer. Then, the vehicle releases the load in the outbound buffer. Meanwhile, the lift moves from its dwell point to the designated tier.

(2) If the lift reaches the designated tier and the retrieval load is in the outbound buffer, the lift picks up the load. If either the lift or the retrieved load reaches the outbound buffer first, it has to wait for the other. After the lift has picked up the load, it moves to the I/O point and releases the load.

If the storage position of the retrieval load is at the ground-floor tier, the lift should only move to the ground-floor tier, picks up the load from the outbound buffer of the ground-floor tier and then releases the load at the I/O point.

For both the storage and retrieval transactions, we call the operations of step (1) that are related to the lift the "vertical task", and those related to the vehicle the "horizontal task". We call the operations of step (2) the "remaining task". Under the parallel processing policy, the lift carries out the vertical task, while meanwhile, the vehicle carries out the horizontal task. After the completion of these two tasks, either the lift picks up the load and carries out the remaining task for a retrieval transaction, or the vehicle picks up the load and carries out the remaining task for a storage transaction.

Since the inbound/outbound buffers are the first storage locations of each aisle, they can hold the same number of loads as other storage locations, i.e., one load. To avoid the waiting of the vehicle for the buffers, we assume that the state of the vehicle is determined by the state of its buffers. Namely, the vehicle is unavailable for storage transaction if its inbound buffer is occupied, and unavailable for retrieval transaction if its outbound buffer is occupied.

In the system described above, the transactions to be performed at the first tier need the lift. This study also considers the system that the transactions to be performed at the first tier do not need the lift, i.e., the vehicle transports the loads into and out of the ground-floor tier.

ACCEPTED MANUSCRIPT

3.1. ***Main notations and assumptions***

We define main notations used throughout the paper as follows:

A: Width of the tier, in the number of storage positions (see Fig.1(b)).

T: Height of the rack, in the number of tiers. Because we investigate one basic unit of the tier-captive AVS/RS, the system examined in this paper consists of one lift and T vehicles.

C: Storage capacity, expressed in the number of storage positions served by one lift. We have $C = 2AT$.

h: Height of one tier (m).

w: Width of one storage position (m).

l: Length of one storage position (m).

v_h: Maximum velocity of the vehicle (m/s).

v_v: Maximum velocity of the lift (m/s).

a_h: Acceleration/deceleration of the vehicle (m/s^2).

a_v: Acceleration/deceleration of the lift (m/s^2).

λ_s, λ_r: Arrival rates of storage and retrieval transactions (per second). We assume that the arrival of transactions follows a Poisson distribution, so the average inter-arrival time of storage/retrieval transactions is λ_s^{-1}, and λ_r^{-1}, respectively.

c_h: Time for picking up or releasing a load by the vehicle (s).

c_v: Time for picking up or releasing a load by the lift (s). Without loss of generality, we assume that both c_h and c_v are constant.

T_r: System response time for retrieval transactions (s).

W_r: Expected waiting time of retrieval transactions at external queue of the system.

ρ_l: Utilization of the lift.

ρ_v: Utilization of the vehicles.

We make the following assumptions for the tier-captive autonomous vehicle storage/retrieval systems:

1. We use the random storage policy, based on a space-conserving consideration (Heragu (2008)). Under this policy, the storage load will be stored in any tier with the same probability $1/T$, and each empty location has the same probability to be filled by the storage load.

ACCEPTED MANUSCRIPT

2. We only consider retrieval transactions in this study. This is based on the following observations. First, retrieval transactions represent the most critical activities of the system, as they directly influence the service level of the warehouse. Second, our model can be easily extended to the case of storage transactions.
3. The lift and the vehicles follow a Point-of-Service-Completion (POSC) dwell point policy. Since we only consider retrieval transactions, the lift will dwell at the I/O point of the rack, and the vehicle will dwell at the outbound buffer at its storage tier.
4. The lift and the vehicles all follow a First-Come-First-Served (FCFS) service rule.
5. We only consider single-command cycles.

3.2. ***Components of transaction cycle times related with the lift and the vehicle***

Since we only consider retrieval transactions in this study, we calculate the components of the retrieval transaction cycle time related to the lift and the vehicle. We call the retrieval transaction the customer of the system, and the retrieval transaction carried out at tier t as a class t customer, $t = 1, 2, \cdots, T$. As a result of the random storage policy, the probability of retrieving a load from tier t is $P(t) = 1/T$, and the vehicle in tier t faces customer demand with a rate of $\lambda_r^t = \lambda_r/T$.

We index the storage locations in the aisle by distance to the outbound buffer, and denote the location of the retrieval load in the aisle by a. Then, the probability of the retrieval load being in storage position a is:

$$P(a) = \frac{1}{A}, a = 1, 2, \cdots, A.$$

Following the POSC dwell point policy, the vehicle will dwell nearby the outbound buffer. To retrieve load a, the vehicle will move from the outbound buffer to the location of load a, pick it up, then return to the outbound buffer. The horizontal travel distance for the vehicle to move from its dwell point to the location of the retrieval load a is

$$H(a) = aw.$$

Considering the effect of acceleration and deceleration on the movement of the vehicle, we obtain the service time of the horizontal task, corresponding to the vehicle moves to the retrieval

ACCEPTED MANUSCRIPT

location, picks up the load, drives to the outbound buffer and releases it in the outbound buffer

$$T_v(a) = \begin{cases} 2 * [2v_h/a_h + (H(a) - v_h^2/a_h)/v_h + c_h], & H(a) > v_h^2/a_h \\ 2 * [2\sqrt{\frac{H(a)}{a_h}} + c_h], & H(a) \leq v_h^2/a_h. \end{cases} \quad (1)$$

The vehicle will reach its maximum velocity v_h in the first case of Eq.(1). In this case, the one way travel time includes three parts: the accelerating time $\frac{v_h}{a_h}$, the rapid travel time $(H(a) - v_h^2/a_h)/v_h$ and the decelerating time $\frac{v_h}{a_h}$. The vehicle cannot reach v_h in the second case. Therefore, the one way travel time includes two parts: the accelerating time $\sqrt{\frac{H(a)}{a_h}}$ and the decelerating time $\sqrt{\frac{H(a)}{a_h}}$.

In this study, we model all service time distributions as general Coxian distributions. This allows analytical tractability, while such distributions can be fit to two moments of any distribution (see Jia and Heragu(2009)). For the approximation of a general service time distribution by a Coxian distribution, one can refer to Appendix **D**. We denote the mean value and the squared coefficient of variation *scv* of $T_v(a)$ by τ_v and cv_v^2, respectively. We obtain τ_v by Eq.(2),

$$\tau_v = \sum_{a=1}^{A} P(a) T_v(a). \quad (2)$$

The retrieval transaction cycle time related to the lift can be divided into two parts (one part is the vertical task and the other is the remaining task). The first part corresponds to the vertical movement from the dwell point of the lift, i.e., the I/O point, to the designated tier t, the second part corresponds to the lift picks up the retrieval load from the outbound buffer, moves from tier t to the I/O point and then releases the load. Both parts have identical travel times, and the one-way travel distance is

$$V(t) = (t-1)h.$$

Considering the effect of acceleration and deceleration on the movement of the lift, the service time of the vertical task, corresponding to the movement from the I/O to the retrieval tier, equals

$$T_{l_1}(t) = \begin{cases} 2v_v/a_v + (V(t) - v_v^2/a_v)/v_v, & V(t) > v_v^2/a_v \\ 2\sqrt{\frac{V(t)}{a_v}}, & V(t) \leq v_v^2/a_v. \end{cases} \quad (3)$$

The lift will reach its maximum velocity v_v in the first case of Eq.(3), and can not reach v_v in the second case. The deviation of Eq.(1) and Eq.(3) are given in Appendix **B**.

We approximate the distribution of $T_{l_1}(t)$ based on its first two moments, its mean value τ_{l_1} and its *scv* $cv_{l_1}^2$, by a phase-type Coxian distribution. We obtain τ_{l_1} by Eq.(4)

$$\tau_{l_1} = \sum_{t=1}^{T} P(t) T_{l_1}(t). \tag{4}$$

Since we adopt the POSC dwell point policy for the lift, the service time of the remaining task is $T_{l_2}(t) = T_{l_1}(t) + 2 * c_v$. Thereby, we can get the mean value and the *scv* of $T_{l_2}(t)$, denoted by τ_{l_2} and $cv_{l_2}^2$, respectively.

4. Fork-Join Queuing Network for Tier-Captive AVS/RSs

In this section, we model the operational process of the tier-captive AVS/RSs under the parallel processing policy as a fork-join queueing network. Then, we develop an approximation method to estimate the performance measures of the system, since the fork-join queueing network is non-product form, because both the vehicle service time and the lift service time are generally distributed.

Fig.2 shows the fork-join queueing network formulated for the tier-captive AVS/RSs. In this queueing network, the retrieval transactions arrive at the fork node with arrival rate λ_r, leading to arrival rates at different tiers $\lambda_r^1, \lambda_r^2, \cdots, \lambda_r^T$. A retrieval request for the t-th tier simultaneously requests the lift and the vehicle at the t-th tier. This operation corresponds to the fork node where the transaction is split into a horizontal task served by the t-th vehicle and a vertical task served by the lift. The finished horizontal task waits in the join queue of the horizontal tasks, denoted by Q_h^J, and the finished vertical task waits in the join queue of the vertical tasks, denoted by Q_v^J. After the completion of these two tasks, they depart their join queues and join at the join node. Then, the transaction goes to the station μ_{l_2} where the lift picks up the retrieval load, and then transports it from the designated tier to the I/O point of the system and releases the load. Since the lift will go with the transaction from the fork-join node to the μ_{l_2} node, the transaction gets service at μ_{l_2} node without waiting. Therefore, we model the μ_{l_2} node as a node with an infinite server.

In the queueing network, μ_{v_t} is the service rate of the t-th vehicle for the horizontal task. As a result of the random storage policy, the service rates of all vehicles are the same and we have

$$\mu_{v_t} = \frac{1}{\tau_v}, t = 1, 2, \cdots, T.$$

ACCEPTED MANUSCRIPT

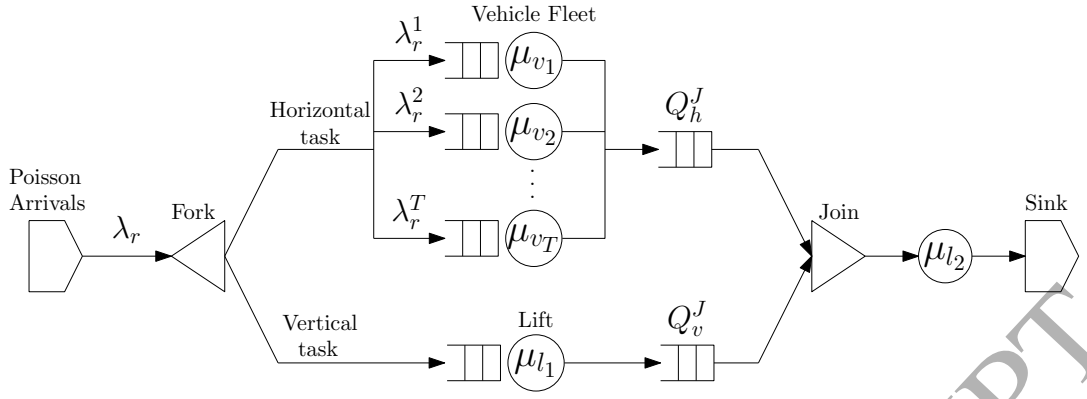


Figure 2: The fork-join queuing network for tier-captive AVS/RSs

To simplify expressions, we define $\mu_v := \mu_{v_t}$.

The components of the retrieval transaction cycle times related with the lift are divided into two parts, T_{l_1} and T_{l_2}. We denote the service rate of the lift for its first part operation by μ_{l_1}. Then we have

$$\mu_{l_1} = \frac{1}{\tau_{l_1}}.$$

Let μ_{l_2} be the service rate of the lift for its second part operation. Then we have

$$\mu_{l_2} = \frac{1}{\tau_{l_2}}.$$

Note that although no lifting is involved at the first tier, the lift is still required to transfer the load into or out the buffer area. Therefore, the horizontal task at the first tier is also included in the fork-join node of Fig.2. For the system that the storage and retrieval transactions at the first tier do not need the lift, we model the service process of the first tier as an independent $M/G/1$ queuing system and the service process of the other tiers as a FJQN (see Appendix **C**).

In order to examine the efficiency of the parallel processing policy, we need to estimate the performance of the fork-join queueing network, including the system response time for retrieval transactions, the expected waiting time of retrieval transactions in the external queue, and the lift utilization and the vehicle utilization. To this end, we develop an approximation method based on the decomposition principle of the fork-join queueing network (see Allen (1990), Bolch (2006)). The principle is that we consider the FJQN as an open queuing network and replace the subnetwork that contains the fork-join construct by a composite load-dependent node. The approximation method includes the following steps:

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

1. The first step is the estimation of the load-dependent service rate of the fork-join node (Section 4.1), including two sub-steps.
   (a) We build a closed queueing network that contains the fork-join construct and define the state variable (see Section 4.1.1).
   (b) We approximate the service process of each node in the closed queueing network by a Coxian-m distribution, derive the state transition matrix, and calculate the system throughput ($\lambda_{fj}(\cdot)$) (see Section 4.1.2).
2. The second step is the estimation of the performance measures of the FJQN (Section 4.2), including two sub-steps.
   (a) We replace the fork-join node by a flow equivalent server (FES) with exponential distributed load-dependent service time, whose service rate is $\mu_{FES}(\cdot) = \lambda_{fj}(\cdot)$. Then, we get a non-product-form open queueing network since the service time of the μ_{l_2} node is generally distributed (see Section 4.2.1).
   (b) We use the Maximum Entropy Method (MEM) to derive the steady state probabilities of the open queueing network. Then, we obtain the performance measures of the FJQN (see Section 4.2.2).

4.1. ***Estimation of load-dependent service rate of the fork-join node***

We estimate the load-dependent service rate of the fork-join node in two steps: First, we build a closed queueing network that only contains the fork-join construct in Section 4.1.1. Second, we approximate the state probabilities of the closed queueing network in Section 4.1.2. With the approximated state probabilities, we calculate the service rate of the fork-join node based on the idea that the throughput of the closed queueing network equals the service rate of the fork-join node.

4.1.1. ***A closed queueing network for the fork-join node***

We build a closed queueing network by short-circuiting the fork-join node (see Fig.3). The number of customers, i.e., retrieval transactions, in this queueing network is constant, K_r. Based on the random storage policy, we assume that the K_r customers are each randomly drawn as a retrieval from one of the T tiers. So, each customer in the closed queueing network represents a request for the t-th vehicle with probability $1/T$.

The aim of building this closed queueing network is that its throughput equals the service rate of the fork-join node. When a customer goes through the Fork node, it will be splitted into a horizontal task that requests the vehicle in designated tier and a vertical task requests for the lift. The joining of the horizontal and the vertical tasks at the join node represents the service completion of a customer.

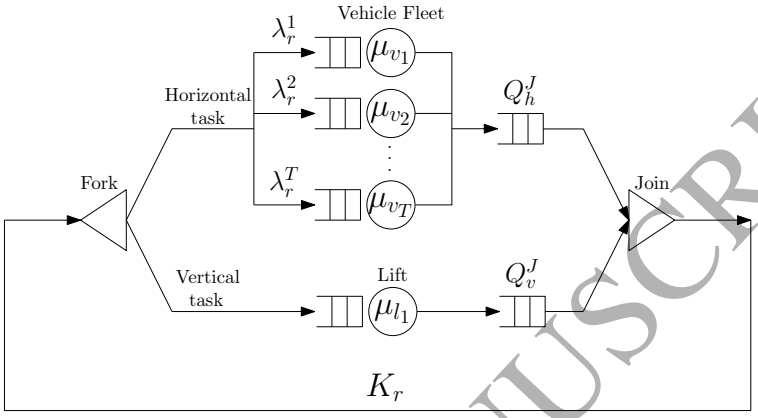


Figure 3: The short-circuited fork-join node

Let Q_h be the number of waiting horizontal tasks in the join queue of the horizontal tasks Q_h^J, and Q_v be the number of waiting vertical tasks in the join queue of the vertical tasks Q_v^J. Then, we define the state variable of this closed queueing network as follows:

$$s_k = (Q_h, Q_v), \text{where } k = Q_v(m+1) + Q_h. \tag{5}$$

Where $m = min(T, K_r)$ is the maximum number of waiting tasks in the join queue of the horizontal tasks Q_h^J.

Since the system examined in this study has one discrete lift with a capacity of one load only, T vehicles, and the outbound buffer of each tier can hold only one load, the state space is

$$Q_v + Q_h < m+1,\ Q_v = 0, 1,\ Q_h = 0, 1, 2, \cdots, m. \tag{6}$$

Note that $Q_v + Q_h = m + 1$ is not possible, since then immediately a horizontal task and a vertical task would be joined.

We note that a successful join of a horizontal task and a vertical task includes two cases: The completion of the vertical task when $Q_h > 1$, which means the lift reaches the target tier to access the waiting load, corresponding to a transition rate μ_{l_1}. The completion of the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

horizontal task when $Q_v = 1$, which means the vehicle reaches the outbound buffer to release the load, corresponding to a transition rate μ_v. Based on this observation, we can calculate the throughput of the closed queueing network with K_r customers, which equals the load-dependent service rate of the fork-join node, by Eq.(7),

$$\lambda_{fj}(K_r) = \sum_{Q_h=0}^{m} \mu_{l_1} \pi(Q_h, 0) + \sum_{Q_h=0}^{m-1} \mu_v \pi(Q_h, 1), \tag{7}$$

where $\pi(s_k)$ is the state probability of state s_k.

4.1.2. ***Approximated state probabilities and throughput of the closed queueing network***

To derive the load-dependent throughput of the closed queueing network $\lambda_{fj}(\cdot)$ from Eq.(7), we need to calculate the state probabilities of the closed queueing network $\pi(s_k)$. To this end, we first approximate the general distributions of the vehicle service time and the lift service time by Coxian distributions (see Altiok (1985), Jia & Heragu (2009), Cai et al. (2014), and Tappia et al. (2016)). Then we derive the transition matrix of s_k, denoted by $\mathbf{Q}$. For the approximation of general distributions by Coxian distributions, one can refer to Appendix **D**.

We denote the number of phases of the Coxian distributions for the vehicle service time and the lift service time by n_v and n_l, respectively, denote the phase-type representation of these two Coxian distributions by $(\boldsymbol{\alpha_v}, \mathbf{T_v})$ and $(\boldsymbol{\alpha_l}, \mathbf{T_l})$. Then, we have $\boldsymbol{\alpha_v} = [1\ 0 \cdots 0]_{1\times n_v}, \boldsymbol{\alpha_l} = [1\ 0 \cdots 0]_{1\times n_l}$, and we obtain the phase transition matrixes $\mathbf{T_v}$ and $\mathbf{T_l}$ by Eq.(D.1) or Eq.(D.3). Additionally, we have an absorbing rate matrix $\mathbf{T_v^o}$ for $\mathbf{T_v}$ and an absorbing rate matrix $\mathbf{T_l^o}$ for $\mathbf{T_l}$. They represent the rates of the customer being absorbed from each phase, and we obtain their expressions by Eq.(D.2) or Eq.(D.6).

After the approximation of the distributions of the vehicle service time and the lift service time, the state variable of the closed queueing network becomes (s_k, i, j), where i, j is the current phase of the service process of the vehicle and the lift ($i = 0, 1, \cdots, n_v, j = 0, 1, \cdots, n_l$), respectively. Since the state variable of the join queues, i.e., (Q_h, Q_v), is independent of i and j, we can use s_k in the transition matrix instead of (s_k, i, j). But we should note that the phase state variable (i, j) is still contained in the transition matrix of the state variable s_k.

We denote the vector of the state probabilities with $Q_v = 0$ as $\boldsymbol{\pi_0}$, the vector of the state probabilities with $Q_v = 1$ as $\boldsymbol{\pi_1}$. Specifically, $\boldsymbol{\pi_0} = [\pi(s_0), \pi(s_1), \cdots, \pi(s_m)]$ and $\boldsymbol{\pi_1} =$

$[\pi(s_{m+1}), \pi(s_{m+2}), \cdots, \pi(s_{2m})]$. Then, we have the following transition matrix of the state variable s_k,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B_{00}} & \mathbf{B_{01}} \\ \mathbf{B_{10}} & \mathbf{B_{11}} \end{bmatrix}. \tag{8}$$

In the transition matrix $\mathbf{Q}$, $\mathbf{B_{00}}$ corresponds to the transition rates from the state of $\boldsymbol{\pi_0}$ to the state of $\boldsymbol{\pi_0}$, and we have

$$\mathbf{B_{00}} = \begin{array}{c} \\ s_0 \\ s_1 \\ \vdots \\ s_{m-1} \\ s_m \end{array} \begin{array}{c} \begin{array}{ccccc} s_0 & s_1 & \cdots & s_{m-1} & s_m \end{array} \\ \begin{pmatrix} \mathbf{T_l} \oplus \mathbf{T_v} & \mathbf{T_v^o}\boldsymbol{\alpha_v} \otimes \mathbf{I_l} & & & \\ \mathbf{I_v} \otimes \mathbf{T_l^o}\boldsymbol{\alpha_l} & \mathbf{T_l} \oplus \mathbf{T_v} - \bar{\mathbf{T}}_\mathbf{l}^\mathbf{o} & \mathbf{T_v^o}\boldsymbol{\alpha_v} \otimes \mathbf{I_l} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I_v} \otimes \mathbf{T_l^o}\boldsymbol{\alpha_l} & \mathbf{T_l} \oplus \mathbf{T_v} - \bar{\mathbf{T}}_\mathbf{l}^\mathbf{o} & \mathbf{T_v^o} \otimes \mathbf{I_l} \\ & & & \mathbf{T_l^o}\boldsymbol{\alpha_l} \otimes \boldsymbol{\alpha_v} & \mathbf{T_l} \end{pmatrix} \end{array},$$

where $\mathbf{I_v}$ is an identity matrix with $n_v \times n_v$ size, $\mathbf{I_l}$ is an identity matrix with $n_l \times n_l$ size, $\oplus$ is Kronecker sum and $\otimes$ is Kronecker product, $\bar{\mathbf{T}}_\mathbf{l}^\mathbf{o} = \mathbf{I_v} \otimes \tilde{\mathbf{T}}_\mathbf{l}^\mathbf{o}$ and

$$\tilde{\mathbf{T}}_\mathbf{l}^\mathbf{o} = \begin{bmatrix} \mu_1^l(1-b_1^l) & & & \\ & \ddots & & \\ & 0 & & \\ & & \ddots & \\ & & & \mu_{n_l}^l \end{bmatrix}_{n_l \times n_l}.$$

$\mathbf{B_{01}}$ corresponds to the transition rates from the state of $\boldsymbol{\pi_0}$ to the state of $\boldsymbol{\pi_1}$, and we have

$$\mathbf{B_{01}} = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{m-1} \\ s_m \end{array} \begin{array}{c} \begin{array}{ccccccc} s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{2m} \end{array} \\ \begin{pmatrix} \mathbf{I_v} \otimes \mathbf{T_l^o} & & & & \\ & \mathbf{I_v} \otimes \mathbf{T_l^o} & & & \\ & & \mathbf{I_v} \otimes \mathbf{T_l^o} & & \\ & & & \ddots & \\ & & & & \mathbf{I_v} \otimes \mathbf{T_l^o} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

ACCEPTED MANUSCRIPT

$\mathbf{B_{10}}$ corresponds to the transition rates from the state of $\boldsymbol{\pi_1}$ to the state of $\boldsymbol{\pi_0}$, and we have

$$\mathbf{B_{10}} = \begin{array}{c} \\ s_{m+1} \\ s_{m+2} \\ s_{m+3} \\ \vdots \\ s_{2m} \end{array} \begin{array}{c} \begin{array}{cccccc} s_0 & s_1 & s_2 & \cdots & s_{m-1} & s_m \end{array} \\ \begin{pmatrix} \mathbf{T_v^o}\boldsymbol{\alpha_v} \otimes \boldsymbol{\alpha_l} & & & & & 0 \\ & \mathbf{T_v^o}\boldsymbol{\alpha_v} \otimes \boldsymbol{\alpha_l} & & & & 0 \\ & & \mathbf{T_v^o}\boldsymbol{\alpha_v} \otimes \boldsymbol{\alpha_l} & & & 0 \\ & & & \ddots & & 0 \\ & & & & \mathbf{T_v^o}\boldsymbol{\alpha_v} \otimes \boldsymbol{\alpha_l} & 0 \end{pmatrix} \end{array}.$$

$\mathbf{B_{11}}$ corresponds to the transition rates from the state of $\boldsymbol{\pi_1}$ to the state of $\boldsymbol{\pi_1}$, and we have

$$\mathbf{B_{11}} = \begin{array}{c} \\ s_{m+1} \\ s_{m+2} \\ s_{m+3} \\ \vdots \\ s_{2m-1} \\ s_{2m} \end{array} \begin{array}{c} \begin{array}{cccccc} s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{2m-1} & s_{2m} \end{array} \\ \begin{pmatrix} \mathbf{T_v} - \tilde{\mathbf{T}}_\mathbf{v}^\mathbf{o} & \mathbf{T_v^o}\boldsymbol{\alpha_v} & & & & \\ & \mathbf{T_v} - \tilde{\mathbf{T}}_\mathbf{v}^\mathbf{o} & \mathbf{T_v^o}\boldsymbol{\alpha_v} & & & \\ & & \mathbf{T_v} - \tilde{\mathbf{T}}_\mathbf{v}^\mathbf{o} & \mathbf{T_v^o}\boldsymbol{\alpha_v} & & \\ & & & \ddots & \ddots & \\ & & & & \mathbf{T_v} - \tilde{\mathbf{T}}_\mathbf{v}^\mathbf{o} & \mathbf{T_v^o}\boldsymbol{\alpha_v} \\ & & & & & \mathbf{T_v} \end{pmatrix} \end{array},$$

where

$$\tilde{\mathbf{T}}_\mathbf{v}^\mathbf{o} = \begin{bmatrix} \mu_1^v(1-b_1^v) & & & \\ & \ddots & & \\ & & 0 & \\ & & & \mu_{n_v}^v \end{bmatrix}_{n_v \times n_v}.$$

In the steady state of $[\boldsymbol{\pi_0}, \boldsymbol{\pi_1}]$, we have,

$$[\boldsymbol{\pi}_0 \quad \boldsymbol{\pi}_1]\mathbf{Q} = 0. \tag{9}$$

Additionally, the normalization condition holds,

$$[\boldsymbol{\pi}_0 \quad \boldsymbol{\pi}_1]\mathbf{e} = 1, \tag{10}$$

where $\mathbf{e}$ is a vector of ones. Combining Eq.(9) and Eq.(10), we can obtain the state probabilities $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_1$. Substituting $\boldsymbol{\pi}_0, \boldsymbol{\pi}_1$ into Eq.(7), we can obtain the load-dependent service rate of the fork-join node $\lambda_{fj}(K_r)$.

ACCEPTED MANUSCRIPT

4.2. ***Estimation of the performance measures of the fork-join queueing network***

To estimate the performance of the tier-captive AVS/RSs under the parallel processing policy, we have two steps: First, we build an open queueing network by replacing the fork-join node by a flow equivalent server with the service rates obtained in Section 4.1 (see Jia & Heragu (2009), Cai et al. (2014), Ekren et al. (2014) and Tappia et al. (2016)). Then, we use the MEM to analyze the open queuing network since it is non-product-form.

4.2.1. ***An open queueing network for the fork-join queueing network***

With the load-dependent service rate of the fork-join node obtained in Section 4.1, we replace the fork-join node of Fig.2 by an FES. We assume that the flow equivalent server has exponentially distributed load-dependent service times, and its service rates are given by $\mu_{FES}(\cdot) = \lambda_{fj}(\cdot)$. Then we obtain an open queueing network depicted in Fig.4.

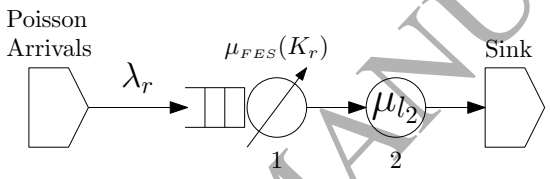


Figure 4: An open queueing network for the fork-join queueing network

The customer arrival process of this open queueing network follows a Poisson distribution with arrival rate λ_r. An arrival customer first gets service of the FES at node 1, whose service rate $\mu_{FES}(K_r)$ depends on the number of customers in it. Then, it gets service at node 2 without waiting, since node 2 contains a server with infinite service capacity. The service time of node 1 is exponentially distributed and load-dependent, and the service time of node 2 follows a general distribution with mean value τ_{l_2} and squared coefficient of variation $cv_{l_2}^2$. So, the open queueing network depicted in Fig.4 is a non-product-form open network. Based on this observation, we apply the MEM for the analysis of the open queueing network (see Bolch (2006)), since this method can get the state probabilities of the open queueing network and the squared coefficients of variation by little computational efforts. The term entropy comes from information theory and is a measure of the uncertainty in the predictability of an event.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

4.2.2. ***Maximum Entropy Method for the open queueing network***

According to the MEM reference (see Kouvatsos (1985)), the approximate state probabilities of the open queueing network are given by Eq.(11),

$$\pi(k_1, k_2) = \pi_1(k_1) * \pi_2(k_2), \tag{11}$$

where k_1, k_2 are the numbers of customers in the first and second node of the network, respectively. $\pi_1(k_1)$ is the marginal probability of the first node, and $\pi_2(k_2)$ is the marginal probability of the second node.

For the first node, we can obtain its marginal state probabilities after recognizing its underlying Markov process,

$$\begin{aligned} \pi_1(0) &= \frac{1}{1 + \sum_{k_1=1}^{+\infty} \prod_{i=1}^{k_1} \frac{\lambda_r}{\mu_{FES}(i)}}, \\ \pi_1(k_1) &= \pi_1(0) * \prod_{i=1}^{k_1} \frac{\lambda_r}{\mu_{FES}(i)}. \end{aligned} \tag{12}$$

For the second node, the MEM provides its marginal probabilities as Eq.(13) (Kouvatsos (1985)),

$$\pi_2(k_2) = \begin{cases} \frac{1}{G_2}, & k_2 = 0 \\ \frac{1}{G_2} \cdot a_2 \cdot b_2^{k_2}, & k_2 > 0, \end{cases} \tag{13}$$

where

$$\begin{aligned} a_2 &= \frac{\rho_2^2}{(1-\rho_2)(\bar{K}_2 - \rho_2)}, \\ b_2 &= \frac{\bar{K}_2 - \rho_2}{\bar{K}_2}, \\ G_2 &= \frac{1}{1-\rho_2}. \end{aligned} \tag{14}$$

In Eq.(14), ρ_2 is the utilization and $\bar{K}_2$ is the mean number of jobs at node 2. We can obtain ρ_2 by $\rho_2 = \lambda_r/\mu_{l_2}$. The MEM approximates $\bar{K}_2$ as a function of the utilization ρ_2, the *scv* of the service time of node 2 cv_2^2 and the *scv* of the inter-arrival times of customers at node 2 cv_{A2}^2.

$$\bar{K}_2 = \frac{\rho_2}{2}(1 + \frac{cv_{A2}^2 + \rho_2 cv_2^2}{1-\rho_2}). \tag{15}$$

In Eq.(15), the *scv* of the inter-arrival times of customers at node 2, i.e., cv_{A2}^2, is computed by the following procedures,

$$cv_{D1}^2 = \rho_1(1-\rho_1) + (1-\rho_1)cv_{A1}^2 + \rho_1^2 cv_1^2, \tag{16}$$

ACCEPTED MANUSCRIPT

$$c_{1,2}^2 = 1 + p_{1,2}(c_{D1}^2 - 1), \tag{17}$$

$$c_{A2}^2 = -1 + [\frac{\lambda_r p_{0,2}}{\lambda_2 \cdot (c_{0,2}^2 + 1)} + \frac{\lambda_1 p_{1,2}}{\lambda_2 \cdot (c_{1,2}^2 + 1)}]^{-1}. \tag{18}$$

In Eq.(16), cv_{D1}^2 is the *scv* of inter-departure times of customers at node 1, cv_{A1}^2 is the *scv* of inter-arrival times of customers at node 1 and cv_1^2 is the *scv* of service times at node 1. Since the customer arrival process follows a Poisson distribution and the service time of node 1 is exponential distributed and load-dependent, we have $cv_{A1}^2 = 1$ and $cv_1^2 = 1$. In Eq.(17), $p_{1,2}$ is the routing probability of a customer from node 1 to node 2 and $p_{1,2} = 1$. In Eq.(18), $p_{0,1}$ and $p_{0,2}$ are separately the routing probabilities of a customer from outside to node 1 and 2, and $p_{0,1} = 1, p_{0,2} = 0$. λ_1 and λ_2 are the arrival rates of customers at node 1 and 2, respectively. We have $\lambda_1 = \lambda_r p_{0,1}$ and $\lambda_2 = \lambda_1 p_{1,2}$.

Based on the observation above, we obtain the *scv* of the inter-arrival times of customers at node 2 $c_{A2}^2 = 1$. Further, we can obtain the marginal probabilities of node 2 by Eq.(13).

With the marginal probabilities of node 1 and node 2, we can obtain the state probabilities of the open queueing network, i.e., $\pi(k_1, k_2)$, by Eq.(11). Then, we can calculate the mean service rate of node 1 by Eq.(19).

$$\bar{\mu}_1 = \sum_{k_1=1}^{+\infty} \pi_1(k_1) \cdot \mu_{FES}(k_1). \tag{19}$$

The expected number of customers at node 1 and node 2 can be calculated by Eq.(20) and Eq.(21), respectively.

$$\bar{K}_1 = \sum_{k_1=1}^{+\infty} k_1 \cdot \pi_1(k_1), \tag{20}$$

$$\bar{K}_2 = \sum_{k_2=1}^{+\infty} k_2 \cdot \pi_2(k_2). \tag{21}$$

According to Little's Law, the system response time is given by Eq.(22),

$$T_r = \frac{\bar{K}_1}{\lambda_r} + \frac{1}{\mu_{l_2}}. \tag{22}$$

The expected waiting time of retrieval transactions in external queue can be calculated by Eq.(23),

$$W_r = \frac{\bar{K}_1}{\lambda_r} - \frac{1}{\bar{\mu}_1}. \tag{23}$$

ACCEPTED MANUSCRIPT

The expected external queue length is calculated by Eq.(24),

$$L_{eq} = \lambda_r W_r. \tag{24}$$

Since we only consider retrieval transactions and each vehicle faces a demand with $\frac{\lambda_r}{T}$ arrival rate, we can calculate the vehicle utilization by Eq.(25),

$$\rho_v = \frac{\lambda_r}{T \cdot \mu_v}. \tag{25}$$

We obtain the lift utilization by Eq.(26),

$$\rho_l = 1 - \pi_1(0) \cdot \pi_2(0). \tag{26}$$

5. Analytical Model Validation and Numerical Experiments

In Section 5.1, the fork-join queueing network formulated in this study is validated with simulation. Then, we carry out a series of numerical experiments to compare the performance of sequential and parallel processing policies in Section 5.2. In Section 5.3, we investigate a real case.

5.1. Validation with simulation

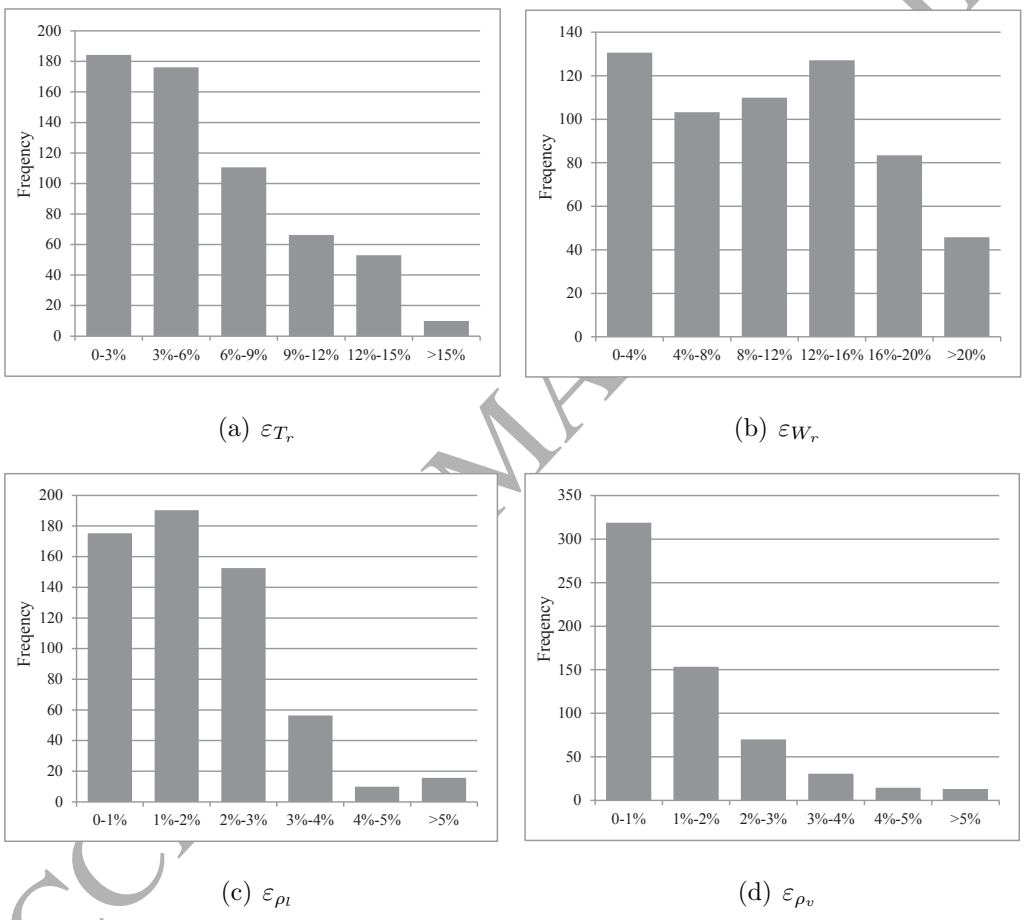
We develop a simulation model based on *Arena*(Version 14.0). Appendix **E** presents the details of simulation models, including the main events and processes, the process flowchart and the simulation validation results. Table 1 presents the system scenarios in simulation model. We consider a tier-captive AVS/RS (with one lift and T vehicles) used for storing standard euro-size totes of $60\times40\times40cm$ ($l\times w\times h$) net, or $60\times50\times80cm$ gross. 6 scenarios are examined based on the variation of T and A. Other assumptions are the same as the analytical models, i.e., random storage policy, POSC dwell point policy for the lift and vehicles and only retrieval transactions are considered. To validate the analytical models under different resource utilizations, the retrieval transaction arrival rate takes 6 levels, $\lambda_r = 50, 75, 100, 125, 150, 175, 200$ per hour.

For each examined case, first a warm-up period of 100 hours is run to eliminate the initial bias, followed by 100 replications of 1000 hours simulation. This leads to a 95% confidence interval where the half-width is less than 2% of the average. We take four performance metrics to validate the analytical models: the system response time for retrieval transactions T_r, the expected waiting time of retrieval transactions in the external queue W_r, the lift utilization ρ_l

Table 1: System scenarios to be examined in simulation model

S	1	2	3	4	5	6
T	5	6	7	8	9	10
A	35	42	50	58	65	72
C	350	504	700	928	1170	1440

[1] Notes: S means system scenarios. $l = 0.6m, w = 0.5m, h = 1.2m, v_v = 4m/s, v_h = 2m/s, a_v = 3m/s^2, a_h = 1m/s^2, c_v = 3s, c_h = 2s$



(a) ε_{T_r} (b) ε_{W_r}

(c) ε_{ρ_l} (d) ε_{ρ_v}

Figure 5: Relative errors of the analytical results to the simulation results

and the vehicle utilization ρ_v. The analytical and simulation results are denoted by R_a, R_s, respectively. The accuracy of analytical models is measured by absolute relative error ε (Eq.(27)).

$$\varepsilon = \frac{\mid R_a - R_s \mid}{R_s} \times 100\%. \tag{27}$$

We present the simulation results, analytical results and relative error in Table 5 and Table

ACCEPTED MANUSCRIPT

6. The average relative error of analytical results to simulation results is $\varepsilon_{T_r} = 4.91\%$, $\varepsilon_{W_r} = 11.77\%$, $\varepsilon_{\rho_l} = 1.82\%$ and $\varepsilon_{\rho_v} = 0.96\%$. The frequency of the relative error is presented in Fig.5.

The simulation validation results show that the analytical models can accurately estimate the system response time for retrieval transactions, the expected waiting time in the external queue and the utilizations of the lift and vehicles.

5.2. Comparison of sequential and parallel processing policies

The motivation of proposing the parallel processing policy is that it may outperform the sequential processing policy for tier-captive AVS/RSs. We carry out numerical experiments to compare the performance of sequential and parallel processing policies. Based on the simulation validation results, we use the fork-join queueing network as the performance estimation tool of parallel processing policy. For sequential processing policy, we choose the open queueing network formulated by Marchet et al. (2012) as the performance estimation tool, which is identical to Fig.2, except that the lift station with processing rate μ_{l_1} precedes the tiers station with parallel tiers. To compare the system response time of retrieval transactions for sequential and parallel processing policies in more detail, we vary T and A, i.e., T ranges from 6 to 13 and $\frac{A}{T}$ takes the values 5,7 and 9. The retrieval transaction arrival rate varies from 50 to 200 per hour with a stepsize of 10. We divide the scenarios and their results into two parts: small and large systems. The small systems correspond to $T = 6, 7, 8, 9$ and $\frac{A}{T} = 5, 7, 9$ and the large systems correspond to $T = 10, 11, 12, 13$ and $\frac{A}{T} = 5, 7, 9$ (Table 2).

For each system scenario, we calculate the average system response time improvement percentage of the parallel processing policy over the sequential processing policy by Eq (28),

$$I = \frac{1}{16} \sum_{\lambda_r=50}^{200} \frac{T_r^s - T_r^p}{T_r^s} \cdot 100\%, \tag{28}$$

where T_r^s and T_r^p are the system response time for retrieval transactions under sequential and parallel processing policies, respectively.

The system response times of retrieval transactions for small and large systems are presented in Fig.6 and Fig.7, respectively. The average improvement percentages I are presented in Table 3.

Table 3 shows that, in small systems, the parallel processing policy always outperforms the sequential processing policy in terms of system response time for retrieval transactions.

ACCEPTED MANUSCRIPT

Table 2: System scenarios for the comparison of sequential and parallel processing policies

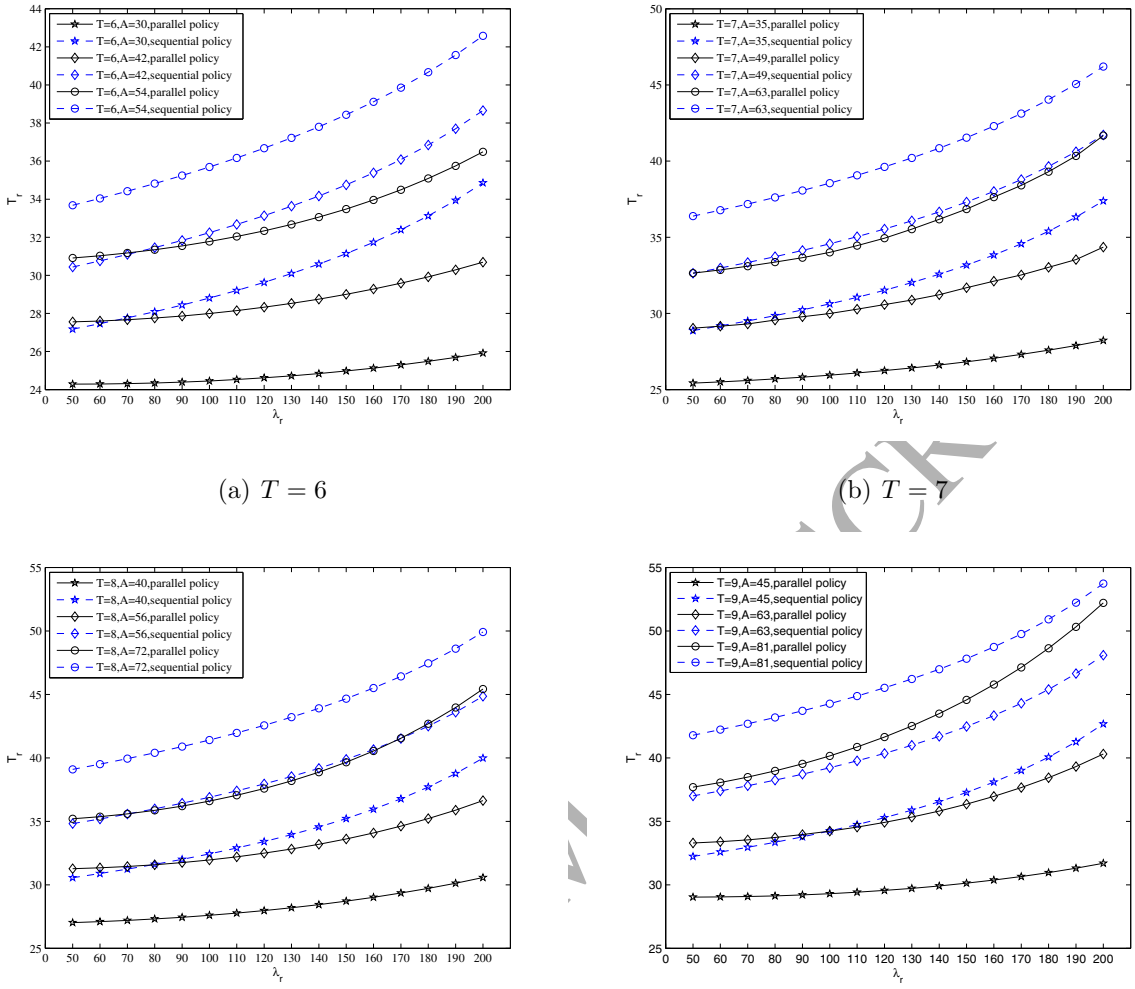
Small	1	2	3	4	5	6	7	8	9	10	11	12
T	6			7			8			9		
A	30	42	54	35	49	63	40	56	72	45	63	91
$\frac{A}{T}$	5	7	9	5	7	9	5	7	9	5	7	9
C	360	504	648	490	686	882	640	896	1152	810	1134	1638
Large	13	14	15	16	17	18	19	20	21	22	23	24
T	10			11			12			13		
A	50	70	90	55	77	99	60	84	108	65	91	117
$\frac{A}{T}$	5	7	9	5	7	9	5	7	9	5	7	9
C	1000	1400	1800	1210	1694	2178	1440	2016	2592	1690	2366	3042

For both small and large systems, the average improvement percentage I decreases with both the rack height T and the aisle length A. In systems with $T \leq 10$, the parallel processing policy outperforms the sequential processing policy and the minimum average improvement percentage is 5.51%.

Table 3: Average improvement percentage of the parallel processing policy over the sequential processing policy

S	1	2	3	4	5	6	7	8	9	10	11	12
$I(\%)$	17.65	14.90	11.72	17.20	14.61	11.14	16.93	14.45	10.83	16.63	13.42	7.51
S	13	14	15	16	17	18	19	20	21	22	23	24
$I(\%)$	15.09	11.55	5.51	14.32	10.41	4.35	13.81	-1.02	-25.88	12.15	-5.61	-44.43

For large systems, we have: Comparing with small size systems, the performance of parallel processing policy becomes poorer in large size systems. In systems with $T > 10$ and $A/T \geq 7$, there exists an intersection point between the curves of the sequential processing policy and the parallel processing policy, denoted by $(\hat{\lambda}_r, \hat{T}_r)$ (For scenario 17, $\hat{\lambda}_r$ is larger than 200 per hour). We call $\hat{\lambda}_r$ as the critical retrieval transaction arrival rate. When $\lambda_r < \hat{\lambda}_r$, the parallel processing policy outperforms the sequential processing policy. The situation reverses when $\lambda_r > \hat{\lambda}_r$. Moreover, $\hat{\lambda}_r$ decreases with both the rack height T and the aisle length A. Although it might seem odd that the sequential policy outperforms the parallel policy for large systems



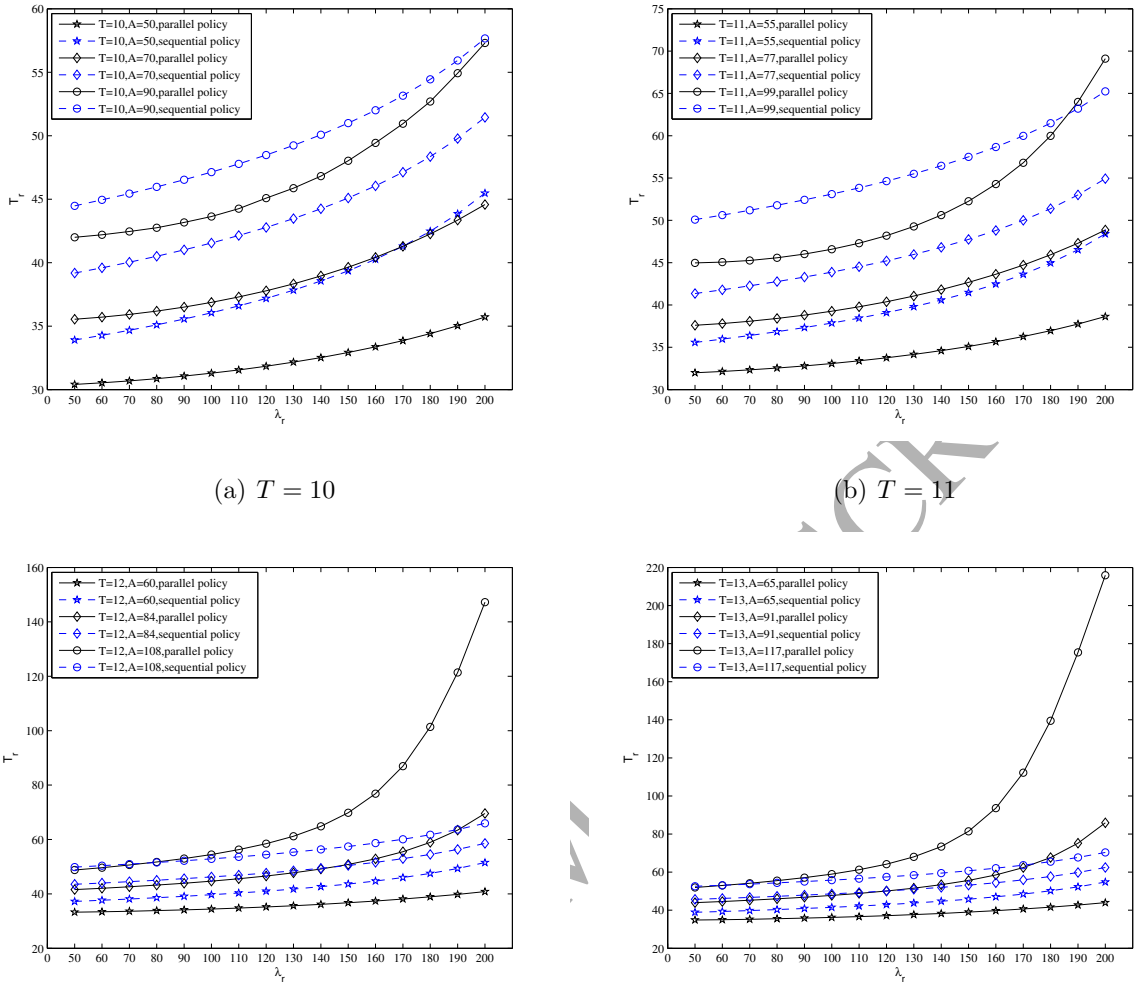
(a) $T = 6$ (b) $T = 7$

(c) $T = 8$ (d) $T = 9$

Figure 6: Comparison of parallel and sequential processing policies in small systems

with large transaction arrival rates, it can be explained by considering joint processing and waiting times separately.

The parallel processing policy reduces the total processing time, since it allows the lift and the vehicle to operate simultaneously. Specifically, the total processing time under the parallel processing policy is the maximum of the lift processing time and the vehicle processing time, while it is the sum of these two time under the sequential processing policy. However, for large systems with large transaction arrival rates, the parallel processing policy increases waiting time due to a heavier work load for the lift, in addition, the lift may also have to wait for the loads. In systems with a low rack, the capacity of the lift is sufficient such that the increase of waiting



(a) $T = 10$ (b) $T = 11$

(c) $T = 12$ (d) $T = 13$

Figure 7: Comparison of parallel and sequential processing policies in large systems

time is dominated by the reduction of total processing time. The parallel processing policy therefore outperforms the sequential processing policy in terms of system response time for retrieval transactions. However, in systems with a tall rack and ($\lambda_r > \hat{\lambda}_r$), loads have to queue at all tiers for the common lift and the reduction of the total operational time cannot offset the increase of the waiting time. So, the parallel processing policy outperforms the sequential processing policy when $\lambda_r \leq \hat{\lambda}_r$, and the situation reverses when $\lambda_r > \hat{\lambda}_r$.

5.3. Investigation of a real case

In this section, we investigate the efficiency of both sequential and parallel processing policies in a real case (an OSR shuttle system of KNAPP). The system consists of 10 single-deep storage

ACCEPTED MANUSCRIPT

tiers and each tier has a tier-captive shuttle that transports loads horizontally. The storage capacity of one tier is 144 totes. One discrete lift moves loads vertically (including the first tier). The current retrieval transaction arrival rate is 118 per hour. Other system parameters are presented in Table 4.

Table 4: System parameters of an OSR case

l	w	h	v_v	a_v	v_h	a_h	c_h	c_v
$0.65m$	$0.35m$	$0.67m$	$5m/s$	$7m/s^2$	$2m/s$	$1.25m/s^2$	$5.1s$	$2.4s$

We vary the retrieval transaction arrival rate varies from 50 to 200 per hour with a stepsize of 10. The results are presented in Fig 8. Under the current service load ($\lambda_r = 118$ per hour), the system throughput can be improved by 2.11% by using the parallel processing policy. The critical retrieval transaction arrival rate is $\hat{\lambda}_r = 155$ per hour, the parallel processing policy outperforms the sequential processing policy in terms of system response time when $\lambda_r \leq 155$, the situation reverses when $\lambda_r > 155$. This result coincides with the findings obtained in Section 5.2, i.e., the system should follow the parallel processing policy under a low service load. Otherwise, the sequential processing policy is better.

6. Conclusions and Future Work

This paper models and analyzes tier-captive autonomous vehicle-based storage and retrieval systems (tier-captive AVS/RSs), in which the vehicle can only visit its designated tier. Different from the tier-to-tier autonomous vehicle-based storage and retrieval systems (tier-to-tier AVS/RSs) in which the vehicle can visit all tiers, the tier-captive AVS/RSs are usually applied for handling small size loads, such as totes. In a tier-captive AVS/RS, the vehicle in the aisle processes the horizontal movement of the load, the lift mounted at the periphery of the rack processes the vertical movement of the load. Due to the variability of the vehicle fleet scale and the independence of the lifts and the vehicles, the system can process storage and retrieval transactions with high flexibility and responsiveness.

Current studies assume sequential commissioning of the lift and vehicles. We propose a parallel processing policy for tier-captive AVS/RSs, under which the lift and the vehicles can serve a transaction simultaneously. To investigate the performance of the parallel processing

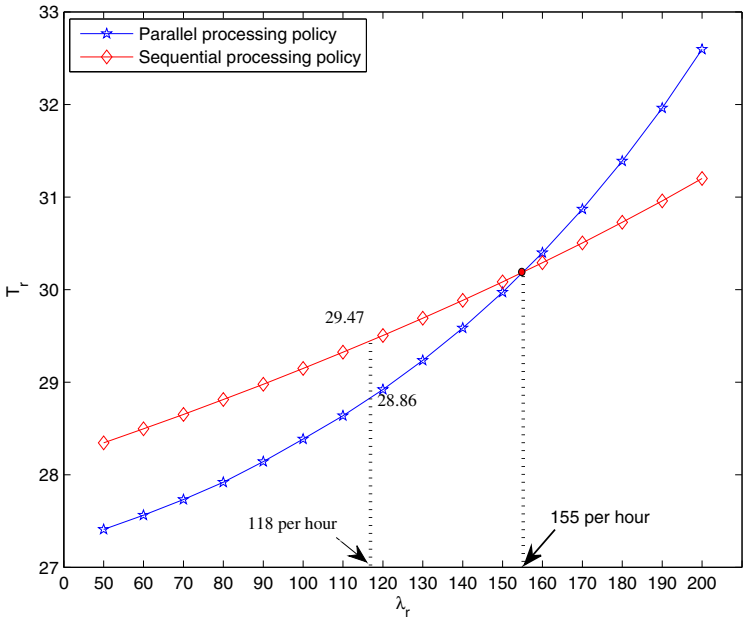


Figure 8: System response times for retrieval transactions under the sequential and parallel processing policies in the OSR case

policy, we formulate a fork-join queueing network in which a transaction will be split into a vertical task served by the lift and a horizontal task served by the vehicle. Then, we develop an approximate method based on the decomposition of fork-join queueing network to estimate the performance of the system, including the system response time for retrieval transactions, the expected waiting time of retrieval transactions in the external queue, the utilizations of the lift and vehicles.

We build simulation models to validate the analytical model. The results show that the analytical model can accurately estimate the performance of tier-captive AVS/RSs under the parallel processing policy. We carry out numerical experiments to compare the system response time of sequential and parallel processing policies. The results show that the parallel processing policy outperforms the sequential processing policy in small size systems. Specifically, the number of tiers is less than 10. In such a system, the average improvement in system response time is at least 5.51%. The advantage of the parallel processing policy decreases with increasing rack height and aisle length.

In large size systems, i.e., the number of tiers is larger than 10 and the ratio of the aisle length to the rack height is larger than 7, we can find a critical point of the retrieval transaction

ACCEPTED MANUSCRIPT

arrival rate. When the retrieval transaction arrival rate is less than this critical level, the tier-captive AVS/RS should follow the parallel processing policy. Otherwise, it should follow the sequential processing policy. We also investigate the performance of both sequential and parallel processing policies in a real case. Under the current retrieval transaction arrival rate, the system throughput can be improved by 2.11% by using parallel processing policy.

In future work, it is interesting to investigate the performance of the parallel processing policy in tier-to-tier AVS/RSs. We can also examine storage transactions and investigate the effect of dwell point policies on the system performance.

Acknowledge

This research is partially supported by the National Natural Science Foundation of China (grant number 71131004), (grant number 71471071), and Chutian Scholarship of Hubei Province of China.

References

Allen, A. (1990). Probability, statistics and queueing theory with computer science applications. *2nd ed.New York:Academic Press*, .

Altiok, T. (1985). On the phase-type approximations of general distributions. *IIE Transactions*, *17*, 110–116.

Bolch, G. (2006). Queuing networks and markov chains, modeling and performance evaluation with computer science applications. *New Jersey: Wiley*, *2nd*.

Cai, X., Heragu, S. S., & Liu, Y. (2014). Modeling and evaluating the avs/rs with tier-to-tier vehicles using a semi-open queueing network. *IIE Transactions*, *46*, 905–927.

Ekren, B. Y., Heragu, S. S., Krishnamurthy, A., & Malmborg, C. J. (2010). Simulation based experimental design to identify factors affecting performance of as/rs. *Computers & Industrial Engineering*, *58*, 175–185.

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

ACCEPTED MANUSCRIPT

Fukunari, M., & Malmborg, C. J. (2008). An efficient cycle time model for autonomous vehicle storage and retieval systems. *International Journal of Production Research*, *46*, 3167–3184.

Fukunari, M., & Malmborg, C. J. (2009). A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research*, *193*, 152–167.

Heragu, S. S. (2008). Facilities design. *Clermont, FL: CRC Press*, .

Heragu, S. S., Cai, X., Krishnumarthy, A., & Malmborg, C. J. (2011). Analytical models for analysis of automated warehouse material handling systems. *International Journal of Production Research*, *49*, 6833–6861.

Hu, Y.-H., Huang, S. Y., Chen, C. Y., Hsu, W. J., Toh, A. C., Loh, C. K., & Song, T. C. (2005). Travel time analysis of a new automated storage and retrieval system. *Computers & Operations Research*, *32*, 1515–1544.

Jia, J., & Heragu, S. S. (2009). Solving semi-open queuing networks. *Operations Research*, *57*, 391–401.

KNAPP (2016). https://www.knapp.com/cms/cms.php. *KNAPP warehouse logistics*, .

Kouvatsos, D. (1985). Maximum entropy methods for general queueing network. *In Pro. Int. Conf. on Modeling Techniques and Tools for Performance Analysis*, *31*, 589–608.

Kuo, P. H., Krishnamurthy, A., & Malmborg, C. J. (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling*, *31*, 2332–2346.

Lerher, T. (2015). Travel time model for double-deep shuttle-based storage and retrieval systems. *International Journal of Production Research*, . doi:`http://dx.doi.org/10.1080/00207543.2015.1061717`.

Lerher, T., Ekren, B. Y., Dukic, G., & Rosi, B. (2015a). Travel time model for shuttle-based storage and retrieval systems. *International Journal of Advanced Manufacturing Technology*, *78*, 1705–1725.

ACCEPTED MANUSCRIPT

Lerher, T., Ekren, B. Y., Sari, Z., & Rosi, B. (2015b). Simulation analysis of shuttle based storage and retrieval systems. *International Journal of Simulation Modeling*, *14*, 48–59.

Malmborg, C. J. (2002). Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, *40*, 1807–1822.

Malmborg, C. J. (2003a). Design optimization models for storage and retrieval systems using rail guided vehicles. *Applied Mathematical Modelling*, *27*, 929–941.

Malmborg, C. J. (2003b). Interleaving dynamics in autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, *41*, 1057–1069.

Marchet, G., Melacini, M., Perotti, S., & Tappia, E. (2012). Analytical model to estimate performance of autonomous vehicle storage and retrieval systems for product totes. *International Journal of Production Research*, *50*, 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*, 4365–4387.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2012). Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. *IIE Transactions*, *44*, 1045–1060.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2015). Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems. *European Journal of Operational Research*, *242*, 72–87.

Tappia, E., Roy, D., Koster, R. D., & Melacini, M. (2016). Modeling, analysis, and design insights for shuttle-based compact storage systems. *Transportation Science in press*, .

Zhang, L., Krishnamurthy, A., Malmborg, C. J., & Heragu, S. S. (2009). Variance-based approximations of transaction waiting times in autonomous vehicle storage and retrieval systems. *Eur. J. Ind. Eng*, *3*, 146–169.

Appendix A

Fig.9 and Fig.10 show the storage and retrieval operations of tier-captive AVS/RSs under parallel processing policy.

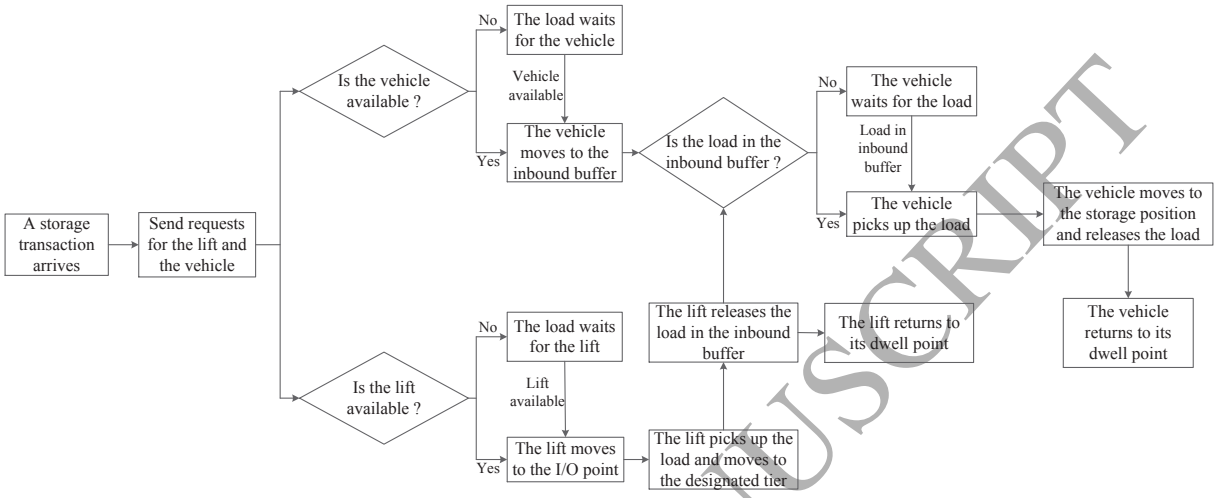


Figure 9: Storage process of tier-captive AVS/RSs under the parallel processing policy

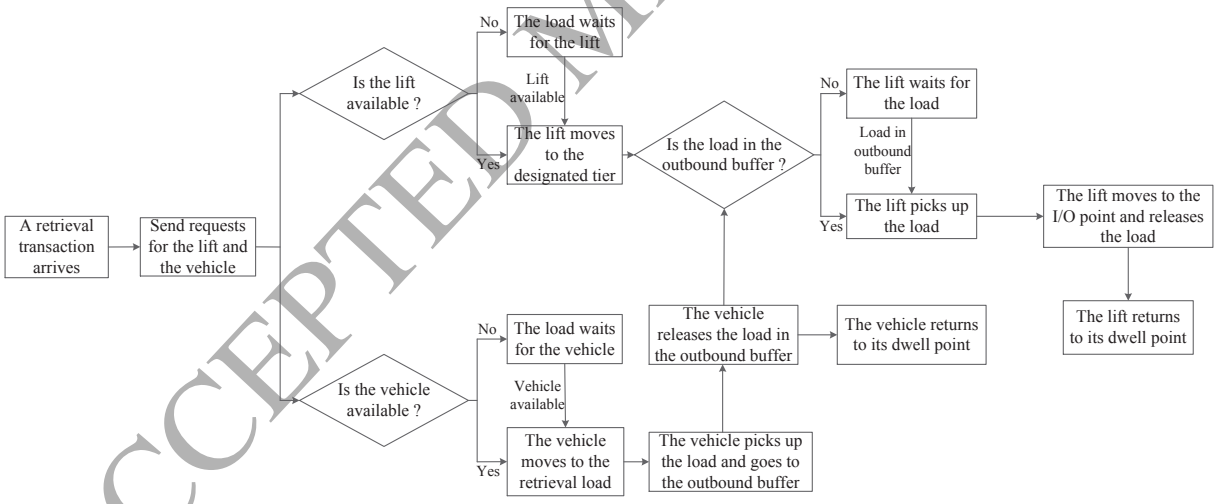


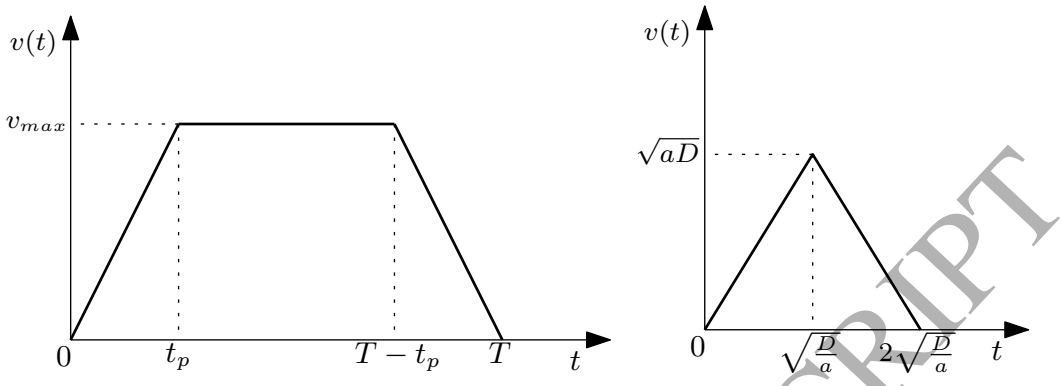
Figure 10: Retrieval process of tier-captive AVS/RSs under the parallel processing policy

Appendix B

We derive Eq.(1) and Eq.(3) by calculating the travel time of a general traveler, considering acceleration and deceleration. Without loss of generality, we assume that the travel distance

ACCEPTED MANUSCRIPT

is D, the maximum velocity of the traveler is v_{max} and the accelerating/decelerating rate is a. Based on whether the traveler can reach v_{max} or not, we have two kinds of velocity-time



(a) $D > v_{max}^2/a$ (b) $D \leq v_{max}^2/a$

Figure 11: Velocity-time relationship based on traveling distance

relationship (see Fig.11). The travel distance for the traveler to accelerates from 0 to v_{max}, or decelerates from v_{max} to 0 is $\frac{v_{max}^2}{2a}$. So, the condition that the traveler can reach its maximum velocity is $D > v_{max}^2/a$.

If $D > v_{max}^2/a$, the traveling period consists of three parts (see Fig.11(a)): accelerating period $[0, t_p)$ where $t_p = \frac{v_{max}}{a}$, rapid traveling period $[t_p, T - t_p)$ where $T = \frac{2v_{max}}{a} + \frac{D - \frac{v_{max}^2}{a}}{v_{max}}$ and decelerating period $[T - t_p, T]$. Therefore, the total traveling time is $T = \frac{2v_{max}}{a} + \frac{D - \frac{v_{max}^2}{a}}{v_{max}}$.

If $D \leq v_{max}^2/a$, the traveling period consists of two parts (see Fig.11(b)): accelerating period $[0, \sqrt{\frac{D}{a}})$ and decelerating period $[\sqrt{\frac{D}{a}}, 2\sqrt{\frac{D}{a}}]$. Therefore, the total traveling time is $2\sqrt{\frac{D}{a}}$.

Appendix C

The following is the queuing networks for tier-captive AVS/RS that the storage and retrieval transactions at the ground-floor tier do not need the lift. The service of the first tier is modeled as a $M/G/1$ queuing system, and the service of the rest part is modeled as a FJQN. We use the approximation method of Allen (1990) to analyze the $M/G/1$ and the approximation method proposed in this study to analyze the FJQN.

Appendix D

Given a general distribution with a mean value τ and a squared coefficient of variation (scv) denoted by cv^2, we can approximate it by a Coxian-2 (C_2) distribution if $cv^2 \geq 0.5$, otherwise, we

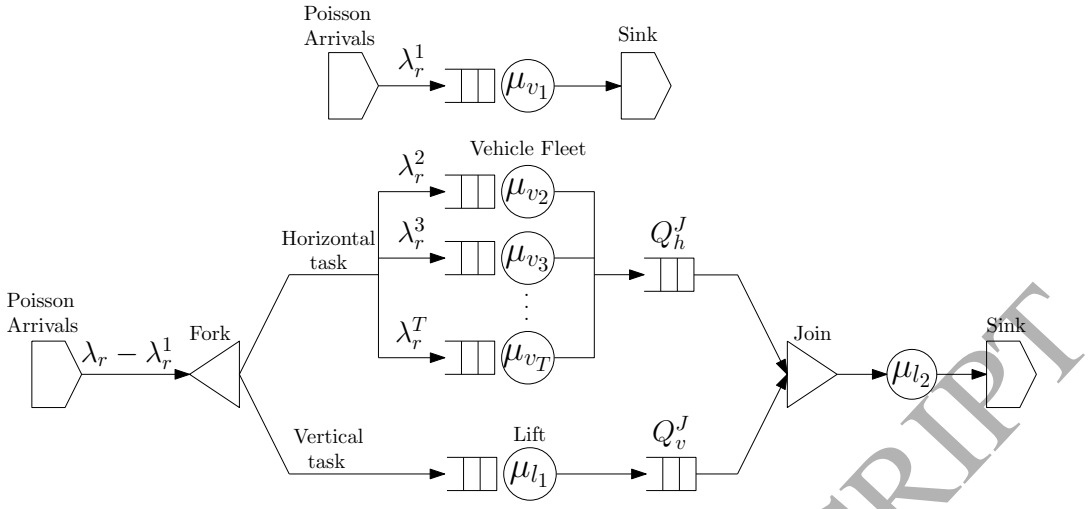


Figure 12: Queuing network for tier-captive AVS/RS that the first tier does not need the lift

use a Coxian-m (C_m) distribution with $m = \lceil \frac{1}{cv^2} \rceil$ to approximate it. If a Coxian-2 distribution is adopted (Fig.13 presents a typical Coxian-2 distribution), a customer starts in the first phase, and then enters the second phase with probability b_1 or gets absorbed with probability $1 - b_1$. The phase-type representation of the Coxian-2 distribution is a pair $(\boldsymbol{\alpha}, \mathbf{T})$. $\boldsymbol{\alpha} = [1\ 0]$ is the initial state probability vector. $\mathbf{T}$ is the phase transition matrix of the underlying absorbing Markov process of the Coxian-2 distribution, and

$$\mathbf{T} = \begin{bmatrix} -\mu_1 & b_1\mu_1 \\ 0 & -\mu_2 \end{bmatrix}, \tag{D.1}$$

where $\mu_1 = 2/\tau, \mu_2 = 1/(\tau * cv^2)$ and $b_1 = 1/(2cv^2)$. Additionally, we have an absorbing rate matrix $\mathbf{T^o}$ representing the rate of the customer being absorbed from each phase, which is shown as

$$\mathbf{T^o} = \begin{bmatrix} \mu_1(1 - b_1) \\ \mu_2 \end{bmatrix}. \tag{D.2}$$

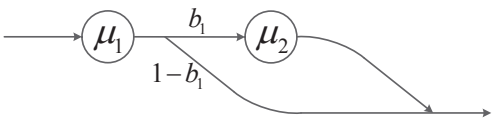


Figure 13: A typical Coxian-2 distribution

If the $cv^2 < 0.5$, a Coxian-m distribution ($m > 2$) is adopted, with $m = \lceil \frac{1}{cv^2} \rceil$ (Fig.14 presents a typical Coxian-m distribution). A customer starts in the first phase, then goes to the second phase with probability b_1 or gets absorbed with probability $1 - b_1$. After the

ACCEPTED MANUSCRIPT

customer finishes the service of the ith phase ($2 \leq i \leq m-1$), it goes to the $(i+1)$th phase with probability one. The customer will get absorbed when it finishes the service of the mth phase.

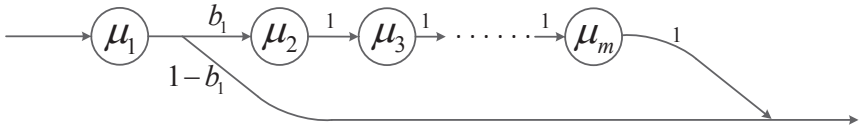


Figure 14: A typical Coxian-m distribution

We can also use a pair $(\boldsymbol{\alpha}, \mathbf{T})$ to represent the Coxian-m distribution. $\boldsymbol{\alpha}$ is the initial state probability vector and $\boldsymbol{\alpha} = [1\ 0\ \cdots\ 0]_{(1\times m)}$, $\mathbf{T}$ is the phase transition matrix of the underlying absorbing Markov process of the Coxian-m distribution, and we have

$$\mathbf{T} = \begin{bmatrix} -\mu_1 & b_1\mu_1 & 0 & 0 & 0 & 0 \\ 0 & -\mu_2 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & -\mu_3 & \mu_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -\mu_{m-1} & \mu_{m-1} \\ 0 & 0 & 0 & 0 & 0 & -\mu_m \end{bmatrix}, \tag{D.3}$$

where

$$b_1 = \frac{2mcv^2 + m - 2 - \sqrt{m^2 + 4 - 4mcv^2}}{2(m-1)(cv^2+1)}, \tag{D.4}$$

and

$$\mu_i = [m - b_1(m-1)]\mu,\ i = 1, 2, \cdots, m. \tag{D.5}$$

The absorbing rate matrix $\mathbf{T^o}$ is

$$\mathbf{T^o} = \begin{bmatrix} \mu_1(1-b_1) \\ 0 \\ \vdots \\ 0 \\ \mu_2 \end{bmatrix}_{m\times 1}. \tag{D.6}$$

Appendix E

The following are the main events and processes in the simulation model:

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

1. Arrival of retrieval transactions: The arrival process at each tier follows a Poisson distribution with arrival rate λ_r/T.
2. A retrieval transaction requests the lift and the target vehicle: Under the parallel processing policy, a retrieval transaction requests the lift and the vehicle at the retrieval tier, simultaneously. The simulation model captures this process by a "split" module, where the retrieval transaction is split into a horizontal task that requests the vehicle at the retrieval tier and a vertical task that requests the lift.
3. Transportation of a retrieval load from its storage position to the outbound buffer: After the vehicle is seized by the horizontal task, the vehicle moves from the entrance of the aisle to the retrieval position, picks up the load, moves with the load to the outbound buffer and unloads it in the buffer area. The service time depends on the storage position of the retrieval load.
4. Movement of lift from I/O to the target tier: After the lift is seized by the vertical task, the lift platform moves from the I/O of the system to the target tier. The lift is modeled as a server; the service time depends on the tier of the retrieval item. The lift is not be released after this operation.
5. Waiting of the lift for the retrieval load or of the load for the lift: If the lift reaches the target tier first, it will wait for the retrieval load. Otherwise, the retrieval load will wait for the lift at the outbound buffer. The simulation model captures this by a "match" modular that contains two queues. The lift waits for the retrieval load in one queue and the retrieval load waits for the lift in another. When both the lift and the retrieval load are in position, they will join into a retrieval transaction.
6. Transportation of the retrieval load from the retrieval tier to the I/O point: The lift loads the retrieval load from the outbound buffer, moves to the I/O point and then unloads the load. The service time depends on the retrieval tier. Note that the lift was already seized in the fourth event.
7. Departure of the retrieval transaction: The retrieval transaction is finished when the lift releases the load at the I/O of the system.

Fig.15 is the flowchart of the simulation process.

Table 5 and Table 6 present the results of the simulation validation. The simulation results (R_s) are the average of 100 replications.

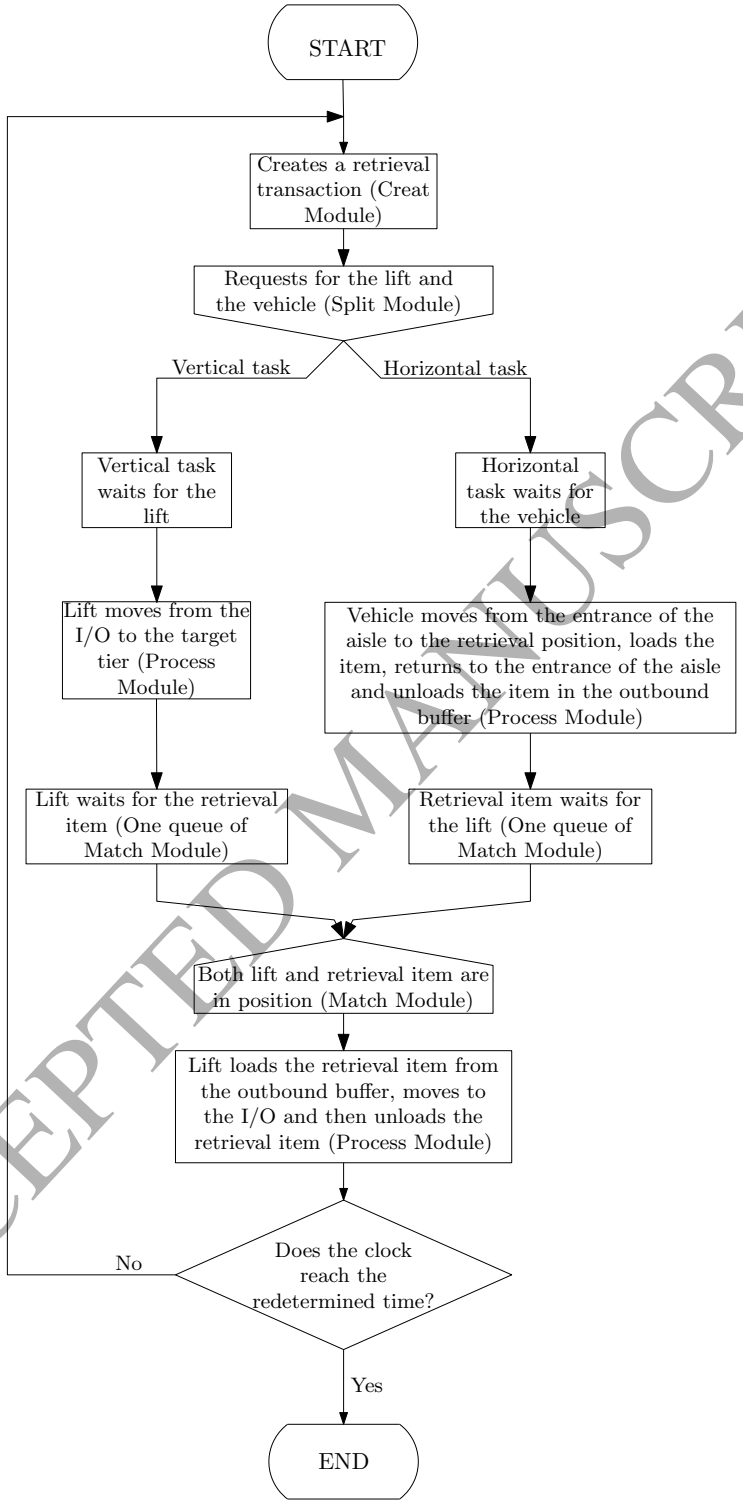


Figure 15: Flowchart of simulation process

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 5: Simulation validation results of scenario one, two and three

S	1			2			3		
	R_a	R_s	$\varepsilon(\%)$	R_a	R_s	$\varepsilon(\%)$	R_a	R_s	$\varepsilon(\%)$
$\lambda_r = 50$									
T_r	25.52	24.31	4.98	27.56	26.45	4.20	29.62	28.99	2.17
W_r	3.44	3.75	8.27	4.09	4.56	10.31	4.82	5.45	11.56
$\rho_l(\%)$	28.35	27.61	2.68	30.29	30.49	0.66	32.81	32.65	0.49
$\rho_v(\%)$	4.14	4.13	0.24	3.86	3.84	0.52	3.70	3.68	0.54
$\lambda = 75$									
T_r	25.62	25.20	1.67	27.71	27.45	0.95	30.02	30.12	0.33
W_r	4.65	5.06	8.10	5.52	6.70	17.61	6.32	7.05	10.35
$\rho_l(\%)$	40.21	40.31	0.25	42.65	42.87	0.51	46.13	45.65	1.05
$\rho_v(\%)$	6.20	6.23	0.48	5.78	5.76	0.35	5.56	5.51	0.91
$\lambda = 100$									
T_r	25.83	26.15	1.22	28.00	28.56	1.96	30.58	31.49	2.89
W_r	5.73	6.51	11.98	6.79	8.20	17.20	7.73	8.65	10.64
$\rho_l(\%)$	50.33	50.68	0.69	53.04	53.64	1.12	57.22	56.82	0.70
$\rho_v(\%)$	8.27	8.31	0.48	7.71	7.67	0.52	7.41	7.39	0.27
$\lambda = 125$									
T_r	26.14	27.24	4.04	28.42	29.86	4.82	31.27	33.01	5.27
W_r	6.74	8.15	17.30	8.01	9.93	19.34	9.34	10.44	10.54
$\rho_l(\%)$	58.96	59.77	1.36	61.77	62.87	1.75	66.38	66.20	0.27
$\rho_v(\%)$	10.34	10.38	0.39	9.64	9.57	0.73	9.26	9.21	0.54
$\lambda = 150$									
T_r	26.57	28.56	6.97	29.00	31.44	7.76	32.18	34.82	7.58
W_r	7.75	9.83	21.16	9.24	11.96	22.74	10.94	12.52	12.62
$\rho_l(\%)$	66.30	67.63	1.97	69.10	70.84	2.46	73.89	74.08	0.26
$\rho_v(\%)$	12.41	12.45	0.32	11.57	11.52	0.43	11.11	11.06	0.45
$\lambda = 175$									
T_r	27.13	30.14	9.99	29.75	33.34	10.77	33.26	37.14	10.45
W_r	8.79	11.84	25.76	10.54	14.10	25.25	12.90	15.28	15.58
$\rho_l(\%)$	72.54	74.38	2.47	75.23	77.51	2.94	80.01	80.50	0.61
$\rho_v(\%)$	14.48	14.51	0.21	13.50	13.42	0.60	12.97	12.91	0.46
$\lambda = 200$									
T_r	27.82	32.13	13.41	30.69	35.78	14.23	34.69	40.03	13.34
W_r	9.90	14.09	29.74	11.95	16.34	26.87	15.44	18.93	18.44
$\rho_l(\%)$	77.81	80.21	2.99	80.33	83.12	3.36	84.92	85.92	1.16
$\rho_v(\%)$	16.54	16.60	0.36	15.42	15.35	0.46	14.82	14.79	0.20

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 6: Simulation validation results of scenario four, five and six

S	4			5			6		
	R_a	R_s	$\varepsilon(\%)$	R_a	R_s	$\varepsilon(\%)$	R_a	R_s	$\varepsilon(\%)$
$\lambda_r = 50$									
T_r	31.75	30.85	2.92	33.78	33.22	1.69	36.06	35.49	1.61
W_r	5.64	6.13	7.99	6.42	6.87	6.55	7.30	7.83	6.77
$\rho_l(\%)$	34.14	34.32	0.52	35.90	36.36	1.27	37.76	38.24	1.26
$\rho_v(\%)$	3.59	3.57	0.56	3.46	3.46	0.00	3.36	3.36	0.00
$\lambda = 75$									
T_r	32.02	32.11	0.28	34.15	34.34	0.55	36.61	37.17	1.51
W_r	7.54	7.99	5.63	8.58	9.08	5.51	9.81	10.23	4.11
$\rho_l(\%)$	47.36	47.68	0.67	49.48	50.06	1.16	51.71	54.83	5.69
$\rho_v(\%)$	5.39	5.35	0.75	5.19	5.19	0.00	5.04	5.03	0.20
$\lambda = 100$									
T_r	32.52	33.47	2.84	34.80	35.80	2.79	37.49	38.43	2.45
W_r	9.27	10.05	7.76	10.56	11.28	6.38	12.15	12.75	4.71
$\rho_l(\%)$	58.16	58.86	1.19	60.43	61.36	1.52	62.82	63.75	1.46
$\rho_v(\%)$	7.18	7.18	0.00	6.93	6.92	0.14	6.72	6.68	0.55
$\lambda = 125$									
T_r	33.27	34.97	4.86	35.76	37.36	4.28	38.75	40.27	3.77
W_r	10.98	12.54	12.44	12.57	14.18	11.35	14.55	15.77	7.74
$\rho_l(\%)$	66.98	68.30	1.93	69.26	70.60	1.90	71.64	73.07	1.96
$\rho_v(\%)$	8.98	8.98	0.00	8.66	8.65	0.12	8.40	8.36	0.48
$\lambda = 150$									
T_r	34.31	36.71	6.54	37.09	39.27	5.55	40.46	42.47	4.73
W_r	12.80	15.30	16.34	14.75	17.53	15.86	17.19	18.99	9.48
$\rho_l(\%)$	74.17	75.83	2.19	76.35	78.06	2.19	78.60	80.43	2.28
$\rho_v(\%)$	10.77	10.75	0.19	10.39	10.36	0.29	10.08	10.07	0.10
$\lambda = 175$									
T_r	35.72	38.96	8.32	38.91	41.74	6.78	42.73	45.30	5.67
W_r	14.86	18.50	19.68	17.27	21.78	20.71	20.25	22.71	10.83
$\rho_l(\%)$	80.01	80.50	0.61	82.02	84.02	2.38	84.06	86.15	2.43
$\rho_v(\%)$	12.57	12.52	0.40	12.12	12.08	0.33	11.76	11.69	0.60
$\lambda = 200$									
T_r	37.60	44.23	14.99	41.37	45.10	8.27	45.77	49.12	6.82
W_r	17.28	22.04	21.60	20.32	25.35	19.84	23.96	27.08	11.52
$\rho_l(\%)$	84.71	88.22	3.98	86.51	88.82	2.60	88.28	90.60	2.56
$\rho_v(\%)$	14.36	14.31	0.35	13.85	13.82	0.22	13.44	13.39	0.37

ACCEPTED MANUSCRIPT