Modelling and Scheduling for Tier-to-Tier Multi-Shuttle Warehousing System

Chengtao Zhu¹, Yaohua Wu²*, GuoZhen Chen³ and ShaSha Wu⁴
School of Control Science and Engineering, Shandong University, Jinan, Shandong, 250061, China
*Corresponding author’s e-mail: 201814510@mail.sdu.edu.cn

Abstract. Tier-to-tier Multi-Shuttle Systems (TT-MSS) become increasingly adopted due to their high flexibility and efficiency. In the Tier-to-tier MSS, shuttles work across different tiers with the help of lifts. How to schedule these devices matters a lot in terms of lowering the expected order cycle time. In this paper, we propose a time sequence task scheduling model considering the movements of shuttles and lifts. The model describes the task scheduling problem between shuttles and lifts, which is generated in a specified time window. An improved genetic algorithm is proposed to solve the objective optimization function in the task scheduling problem. Finally, we illustrate the advantages of the Tier-to-tier MSS by conducting experiments, and the results indicate that with the increase of retrieval tasks, the efficiency of Tier-to-tier MSS is approaching that of TC-MSS with fewer shuttles used.

1. Introduction
In recent years, the Automated Warehouse System (AWS) has been widely promoted due to the rapid development of electronic commerce. Compared with the traditional warehouse, AWS occupies less land, low error rate, higher efficiency and save labor costs. There are already diverse AWS available to meet different storage and sorting needs. The Multi-tier Shuttle System (MSS) is a typical one that is used to store and retrieve products. In a conventional warehousing system, each storage or retrieval task is performed using an aisle-captive stacker crane which moves in horizontal and vertical directions simultaneously. While a typical MSS consists of a series of storage aisles, each of which has an input and output (I/O) point located at one end of the aisle, and lifts responsible for the vertical movement of unit load are located outside the first column of the storage rack, while shuttles are used to transmit the unit loads horizontally. As illustrated in figure 1.
Fig. 1 Overview of a Multi-Shuttle System

On the basis of inheriting the advantages of AVS/RS, MSS has high flexibility that can adapt to the warehousing fluctuation. Therefore, MSS meets the trend of automated warehousing systems for customized storage and small but frequent orders.

MSS can be divided into two types -- tier-captive and tier-to-tier MSS. In tier-captive MSS, each tier has an individual shuttle. While in tier-to-tier MSS, the number of shuttles is typically less than the number of tiers, and shuttles can be moved to the more than one tiers vertically by lifts.

Roodbergen et al. (2009) summarized the literature between 1979 and 2009 and listed the latest techniques for AS/RS design. They found that the optimally organization of the picking sequence is effective in improving the efficiency of the MSS and reducing costs. In view of the high similarity between MSS and AVS/RS and the previous studies mostly focused on the latter one, to outline the research on AVS/RS for reference of MSS is necessary.

AVS/RS was introduced in the United States and Europe around 2000 due to its combination of AGV’s flexibility and AVS/RS’s high-speed accessibility. (Fukunari, 2005). Malmborg (2002) used queuing model to evaluate AVS/RS’s performance. Kuo et al. (2008) proposed a cycle-time model for analyzing the expected cycle-time of AVS/RS under different scenarios, which meets the accuracy requirements of system conceptualization research though the queuing approximation used has some errors. Then they used a non-Poisson queuing model to calculate the storage and retrieval waiting time for shuttle and lift. Zhang et al. (2009) applied the variance approximation to the AVS/RS cycle time model and reduced the error of transaction wait time estimation by about 80%. Marchet et al. (2012) put forward some hypotheses for evaluating various AVS/RS models and summarized both dynamic models and static approaches based on simulation and travel-time modelling-based static approaches. Ekren (2011) proposed the seven classical indicators for evaluating the expected performance of AVS/RS in various design scenarios. Marchet et al. (2013) found that minimizing the number of aisles and maximizing the length of aisles reduced the number of shuttle required for tier-captive AVS/RS by simulations. And the conceptual tools in performance modeling played a significant role in determining storage capacity and guiding rack configuration, which had a significant impact on subsequent research.

There are also some papers on AVS/RS or AS/RS systems using other similar heuristic algorithms. These papers are helpful to understand the advantages and disadvantages of different algorithms, as well as to select the right algorithm to apply to our model.

Brezovnik et al. (2015) adopted a multi-objective ant colony algorithm to solve the scheduling problem of AS/RS. Yang et al. (2015) studied the scheduling problem of multi-shuttle AS/RS and found that the Tabu-Search (TS) algorithm could be better applied to especially large-scale problems than the genetic algorithm. Waurers et al. (2016) applied the heuristic decomposition method in the mini-load AS/RS and found that the optimal solution could only be calculated within a certain
computational time. Jiang and Yang (2017) \cite{13} established an integer programming model for an end-of-aisle multi-vehicle AS/RS retrieval scheduling problem and designed a heuristic algorithm for scheduling analysis. Numerical experiments verified that the performance difference of the algorithm was between 13.99% and 45.75%, which was better than the FCFS rule.

Wang et al. (2015) \cite{14} studied the motion characteristics of shuttles&lifts, and based on this, they proposed a timing sequence model to describe the task operations of MSS system. Later, they (2015) \cite{15} treated the shuttles and lifts as servers in different phases and proposed a two-phase open queueing network model to analyze the performance of MSS. Then the ant colony clustering algorithm was used to solve the scheduling problem. Yang et al. (2018) \cite{16} applied a hybrid plant propagation algorithm (HPPA) to the routing optimization problem of double-loading multi-tier shuttle warehousing system (DMS/WS). Küükyar et al. (2020) \cite{17} simulated and compared the performance of the two types of MSS based on the number of aisles, tiers and bays. The experimental results show that the proposed tier-to-tier MSS has better performance in terms of total investment cost and throughput, and can be used to replace the tier-captive MSS. Wang et al. (2020) \cite{18} established a mixed integer programming model of Tier-captive MSS and proposed several heuristic algorithms to solve its retrieval scheduling problem. They found to obtain an exact solution is impossible when the number of tasks reached double digits in a finite amount of time, and the three heuristic algorithms had their own characteristics when solving large sized problems.

This paper is organized as follows. Section 2 describes the MSS system and analyzes the workflow of a tier-to-tier MSS. In section 3, an objective optimization model is proposed for the task scheduling problems in the MSS. A solution to the model is formulated based on an advanced GA algorithm in Section 4. Section 5 describes experiments and illustrates the results. Finally, Section 6 provides conclusions and provides insights for future study.

2. System description

Tier-to-tier MSS might have much more complex mechanical structure and outbound operations than Tier-captive MSS. figure 2 and figure 3 represent a side view and the flowchart of retrieval tasks of Tier-to-tier MSS respectively.

The biggest difference between Tier-to-tier MSS and Tier-captive MSS is the addition of a lift at the end of the storage racks, which enables the shuttles to move across tiers. And the other parts are same as which used in Tier-captive MSS.

For the MSS, it requires alternating cooperation between the shuttles and the lifts to complete a single retrieval task. While in a tier-to-tier MSS, there might be no shuttle at some tiers which are denoted as shuttle-free tier. When a retrieval task arrives at the shuttle-free tier, the control system needs to assign this task to the nearest idle shuttle at other tiers. Then the assigned shuttle will move to the tail of the aisle where the shuttle lift is located and the lift will take this shuttle to the shuttle-free tier vertically. On arriving at the shuttle-free tier, the shuttle moves horizontally to the retrieval position and the tier turns to shuttle-on tier. Then the shuttles load the target SKU and carry it to the head of the aisle, where the SKU will be handed to the buffer. After that, the shuttle will be released and stop at the first column of the aisle. As soon as the shuttle is assigned a retrieval task, It sends the monitoring system a cargo lift delivery request (CLDR). The CLDR is the estimated earliest arrival time calculated by control system and is inserted into the schedule of cargo lift. Upon receipt of the CLDR, the cargo lift will move to the appropriate tier according to the schedule to load the target SKU from its buffer. Finally, the cargo lift will transfer the SKU to the I/O station and finish this single retrieval task.
To analyze the system performance and task scheduling more easily, we make the following fundamental assumptions.

- Simple operation mode. Only one inbound or outbound order is implemented within a single time window.
- The acceleration/deceleration of shuttle and lift are constant.
- Random storage policy, single-deep storage racks, and input/output (I/O) point at the first tier.
- First-come-first-serve (FCFS) principle. The customer orders share the same priority level. Unit load requests lift (FCFS dispatching).
- Dwell point policy. After completing the previous order, the shuttles and lifts stay in place i.e. the shuttle stays at the first bay of its storage tier, the shuttle stays at the destination tier, and the final position of the cargo lift is the input/output (I/O) station during outbound operations.
- Since we only consider order retrieval tasks in the study, the product shortage is not taken into consideration, i.e. the capacity of buffer is infinite.

3. System modelling

In the last section, we clarify the operation in a parallel retrieval and sequential transfer (PRST)-based MSS. The notation used in this paper are presented in table 1.

According to the FCFS principle, there are two types of lag that may occur during the order fulfillment process and impair the efficiency and performance of the system. The first kind of lag is $t^{SW}$, which is induced when the shuttle has no task to do. We introduce the cross-tier operation mode to share the whole retrieval tasks among all the shuttles. It should be noted that the $t^{SW}$ will not affect the efficiency of cargo lift because of the unlimited buffer in every tier mentioned above. The other stagnation is lift idle time (LIT) and denoted as $t^{lw}$, which occurs if no CLDR is made in the system after the lift finishes the previous task.
Figure 3. Illustration of the Workflow of a Retrieval Task of Tier-to-tier MSS

Let $T_i^r$ denote the requesting time of the task $i$. It is be calculated and sent by the shuttle and is inserted into the schedule of cargo lift. $T_{i-1}^c$ represents the ultimate completion time with respect to the specific task $i-1$ by the cargo lift, assume that $\theta_i$ be the possibility of $t_{lw}^i$. Thus, the accumulative LIT $t_{lw}^i$ can be calculated by equation (1) after $Q$ retrieval tasks are finished:

$$t_{lw}^i = \sum_{i=1}^{Q} \theta_i (T_i^r - T_{i-1}^c)$$  \hspace{1cm} (1)

$$\theta_i = \begin{cases} 1, & T_i^r > T_{i-1}^c \\ 0, & T_i^r \leq T_{i-1}^c \end{cases}$$  \hspace{1cm} (2)

Here, $t_{lw}^i$ represents the total idle time of cargo lift, which is the key parameter that affects the total time of the outbound operation.
Table 1. parameters for tier-to-tier MSS.

| Parameters | Definition |
|------------|------------|
| \( X \)   | Number of storage positions in a single aisle |
| \( Y \)   | Number of tiers in a single aisle |
| \( Z \)   | Number of aisles in each tier |
| \( Q \)   | Total number of retrieval tasks |
| \( Q_y \) | The number of retrieval tasks in Tier \( y \) |
| \( Q_{\text{max}} \) | The maximum number of retrieval tasks in an aisle, \( Q_{\text{max}} = \max_{y \in \{1,2,\ldots,Y\}} Q_y \) |
| \( Y_n \) | Total number of tiers with more than \( n \) retrieval tasks |
| \( Y_i \) | Tier of retrieval task \( i \) |
| \( Y_s^i \) | Tier of shuttle |
| \( Y_l^s \) | Tier of shuttle lift |
| \( Z_i \) | The set of aisle number of storage positions where retrieval tasks in location \( i \). |
| \( a_s \) | Acceleration/deceleration of the shuttle \( (ms^{-2}) \) |
| \( a_l \) | Acceleration/deceleration of the lift \( (ms^{-2}) \) |
| \( V_s \) | The maximum velocity of the shuttle \( (ms^{-1}) \) |
| \( V_l \) | The maximum velocity of the lift \( (ms^{-1}) \) |
| \( t^e \) | Time required for the shuttle to load/unload a SKU \( (s) \) |
| \( t^f \) | Time required for the cargo lift to load/unload a SKU \( (s) \) |
| \( t^g \) | Time required for the shuttle lift to load/unload a shuttle \( (s) \) |
| \( h \) | The height of a single tier \( (m) \) |
| \( a \) | Width of a single storage position \( (m) \) |
| \( L \) | Length of a single aisle \( (m) \) |
| \( T_i^t \) | Starting time for cargo lift handles task \( i \) \( (s) \) |
| \( T_i^r \) | The requesting time for the cargo lift of task \( i \) \( (s) \) |
| \( T_{i-1}^c \) | The completion time for the cargo lift of the previous retrieval task \( (s) \) |
| \( T_{yn} \) | End time for shuttle completes the nth task of tier \( i \) \( (s) \) |
| \( T \) | Planning horizon |
| \( \theta_i \) | Possibility of lift idle time |
| \( t_{sw}^i \) | Shuttle waiting time \( (s) \) |
| \( t_{lw}^i \) | Lift idle time \( (LIT) \) \( (s) \) |
| \( t_{lw}^i \) | The accumulative idle time of cargo lift \( (s) \) |

Due to the cargo lift ultimately should meet the requirements of each customer. To simplify the calculation, the CLDR is regarded as the main line to calculate the total time of outbound operation. As shown in figure 2, the fulfillment of warehousing operation as a whole can be divided into two stages with the cargo lift as the reference frame.

- Stage A. the shuttle travels to retrieve the required SKU (cross-tiers with shuttle lift if necessary), and then returns to the buffer and sends a CLDR. Upon arriving, the shuttle unloads the required SKU into the buffer.
- Stage B. the cargo lift responds to the cargo lift-delivery request (CLDR), travels vertically to the destination tier and loads the outbound SKU from the buffer. After that, it travels to the I/O station and unloads the SKU.

According to Figure 4 and the previous analysis, the entire warehouse operation can be viewed as a pipelined production model. As shown in figure 4, it is assumed that there are four shuttles match the retrieval task. \( A^q_i \) represents the ith\((i=1,2,3,4)\) shuttle transmit the qth retrieval task to the buffer. \( B^q \) represents the cargo lift transmit the qth retrieval task to the I/O station.

The cargo lift task is executed like this. The application for transporting the goods will be submitted to the cargo lift at time \( T_0 \) retrieval task and \( T_1 \) is the earliest CLDR on the schedule. As shown in the
figure, the cargo lift executes the $B^q$ in sequence in the order of the schedule starting from task $B^4$. That is, the outbound operation time (OOT) is equal to the time of the earliest CLDR and the total lift working time (LWT).

![Timing Diagram of cargo lift operation](image)

**Figure 4. Timing Diagram of cargo lift operation**

It is important to notice that the two stages could overlap in time. Because the shuttles send the estimated time of arrival other than send the signal when it arrived at the buffer.

Let $t_i^A$ represents the time spent by task $i$ at stage A, and $t_i^B$ represents the time spent by task $i$ at stage B. According to the production mode of the assembly line, we deduced the total outbound operation time (OOT) as

$$t_{total} = t_1^A + \sum_{i=1}^{Q} t_i^B + t_{hw}$$

($3$)

$t_i^A$ can be calculated as follows:

$$t_i^A = \begin{cases} t_i^{AS}, \\ t_i^{AC}, \end{cases}$$

($4$)

Where $t_i^{AS}$ denotes the CLDR that shuttle at the same tier of required SKU, and $t_i^{AC}$ means the CLDR that shuttle needs to cross tiers to the retrieval tier.

$t_i^{AS}$ and $t_i^{AC}$ can be calculated as follows:

$$t_i^{AS} = 2t_i^{ha} + 2t_i^e$$

($5$)

$$t_i^{AC} = t^L + t_i^{change} + t_i^{hb} + t_i^{ha} + 2t_i^e$$

($6$)

Where $t_i^{ha}$ is the single time of horizontal motion transmission of the shuttle without crossing the aisle and $t_i^e$ indicates the time a shuttle needs to load/unload a single SKU. $t^L$ represents the time for horizontal movement in the original tier of shuttle that need to cross aisles; $t_i^{change}$ is the time spent in cross tiers of shuttle, including the waiting time for the shuttle lift. $t_i^{hb}$ denotes the time of moving to the retrieval position.

In this paper, $t_i^e$ is constant and equals $t^e$ for every retrieval task. Similarly, $t_i^f$ denotes the time of a cargo lift to load/unload a unit and $t_i^g$ indicates the time of a shuttle lift to load/unload a shuttle need to other tiers. These two variables are constant for every task like $t_i^f$, denoted as $t^f$ and $t^g$ respectively.

Here we will list all the remaining constants and derive other variables in consideration of kinematic factors. $a$ corresponds to the width of a storage column; $h$ represents the height of a single tier, $L$ is the entire length of an aisle. $a_s$ and $a_l$ are the acceleration rates of the shuttle and the lift respectively;
while $V_s$ and $V_l$ denote their maximum velocities separately. $(X_i, Y_i)$ represents the position of retrieval task $i$, $Y_i^s$ is the tier of shuttle and $Y_i^{sl}$ is the tier of shuttle lift.

$t_{i}^{\text{change}}$ can be calculated as follows:

$$t_{i}^{\text{change}} = t_{i}^{Y} + 2t_{i}^{R} + t_{i}^{sw}$$

(7)

Where $t_{i}^{Y}$ denotes the pass time for vertical movement of shuttle lift from the tier $Y_i^s$ to the tier $Y_i$ with shuttle, $t_{i}^{sw}$ means the waiting time of shuttle for the shuttle lift at the tier $Y_i^s$ and can be calculated as follows:

$$t_{i}^{sw} = \begin{cases} t_{i}^{q} - t_{i}^{L}, & t_{i}^{q} > t_{i}^{L} \\ 0, & t_{i}^{q} \leq t_{i}^{L} \end{cases}$$

(8)

Where $t_{i}^{q}$ denotes the time of vertical movement of shuttle lift from the tier $Y_i^{sl}$ to the tier $Y_i^s$.

$t_{i}^{B}$ can be calculated as follows:

$$t_{i}^{B} = 2t_{i}^{h} + 2t_{i}^{f}$$

(9)

Where $t_{i}^{h}$ denotes the single-pass time for vertical movement of cargo lift from tier $Y_i$ to I/O station.

We derive the every horizontal and vertical movement times as follows:

$$t_{i}^{ha} = \begin{cases} \frac{2V_s}{a_s} + \left(D_{si} - V_s^2/a_s\right)V_s^{-1}, & D_{si} > V_s^2/a_s \\ 2\left(D_{si}/a_s\right)^{1/2}, & D_{si} \leq V_s^2/a_s \\ D_{si} = a \cdot x_i \end{cases}$$

(10)

$$t_{i}^{L} = \frac{2V_s}{a_s} + \left(L - V_s^2/a_s\right)V_s^{-1}$$

(11)

$$t_{i}^{hb} = \begin{cases} \frac{2V_l}{a_l} + \left(L_s - V_l^2/a_l\right)V_l^{-1}, & L_s > V_l^2/a_l \\ 2\left(L_s/a_l\right)^{1/2}, & L_s \leq V_l^2/a_l \\ L_s = L - a \cdot x_i \end{cases}$$

(12)

$$t_{i}^{Y} = \begin{cases} \frac{2V_l}{a_l} + \left(H_{ls} - V_l^2/a_l\right)V_l^{-1}, & H_{ls} > V_l^2/a_l \\ 2\left(H_{ls}/a_l\right)^{1/2}, & H_{ls} \leq V_l^2/a_l \\ H_{ls} = h \cdot \left|Y_l^s - Y_l\right| \end{cases}$$

(13)

$$t_{i}^{q} = \begin{cases} \frac{2V_l}{a_l} + \left(H_{le} - V_l^2/a_l\right)V_l^{-1}, & H_{le} > V_l^2/a_l \\ 2\left(H_{le}/a_l\right)^{1/2}, & H_{le} \leq V_l^2/a_l \\ H_{le} = h \cdot \left|Y_l^s - Y_{le}\right| \end{cases}$$

(14)

$$t_{i}^{h} = \begin{cases} \frac{2V_l}{a_l} + \left(H - V_l^2/a_l\right)V_l^{-1}, & H > V_l^2/a_l \\ 2\left(H/a_l\right)^{1/2}, & H \leq V_l^2/a_l \\ H = h \cdot \left|Y_l^s - Y_l\right| \end{cases}$$

(15)

Define three Boolean variables to represent the states of lifts and tasks. When task $i$ of lift is located at tier $y$, $b_{x_i y} = 1$. $b_{y in} = 1$ means task $i$ of lift is the nth task of a certain tier, and $b_{zynx} = 1$ represents the retrieval task $n$ of tier $y$ is located at the xth storage position. Otherwise, these variables equal 0.

Objective function:

$$\text{mint}_{\text{total}} = t_{i}^{A} + \sum_{i=1}^{Q} t_{i}^{B} + t_{i}^{lw}$$

(21)

s.t.

$$\sum_{y=1}^{Y} b_{x_i y} = 1, i = 1, 2, \ldots, Q$$

(22)
\[
\sum_{i=1}^{Q} b_{y_{in}} = Y_{n}, n = 1,2,\ldots,Q_{max}
\]  
(23)

\[
\sum_{n=1}^{S_{max}} b_{y_{in}} = 1, i = 1,2,\ldots,Q
\]  
(24)

\[
t_{1} \geq \sum_{n=1}^{y} t_{n}^{h} b_{y_{in}}
\]  
(25)

Constraint (22) specifies that each retrieval task can only appear in one tier. (23) guarantees the total number of the nth retrieval tasks belonging to a certain tier in lift task is equal to the number of tiers with more than n retrieval tasks. (25) means that the time when the first retrieval task starts is later than the time when lift arrives at any tier.

4. Genetic algorithm

The purpose of solving the travel time model of the system is to find the optimal outbound order as we have derived the \( t_{total} \) in forms of kinematic. When the retrieval task reaches a certain scale, the solution space will be too huge to find the optimal solution within a limited time, which is NP-hard problem. It is well known that the small-scale NP problem can be solved by branch and bound, dynamic or integer programming modes, while the problem studied in this paper belongs to a large-scale NP problem. Therefore, we decided to introduce an improved genetic algorithm to solve this problem.

Genetic algorithm is an adaptive mode random search algorithm, which requires less for the objects to be optimized. It does not require objects to be continuous, and does not need to be derivable, with good stability and parallel search mechanism\[19\].

The main idea of GA is creating an initial population with diversity to allow for parallel search threads in the search space, and then repetitively combining the good merits of the individuals via crossovers and optionally mutation steps, and finally applying natural selection for each generation. Although the initial population has a certain importance, the performance of GA does not heavily depend on it.

The task scheduling problem is solved by genetic algorithm, including the steps of setting initial population, fitness function calculation, selection, crossover and mutation, population regeneration.

4.1. Chromosome representation

In the proposed GA, a potential task sequence is represented as a set of parameters known as genes by sequential natural number coding mode, each gene \( r_{i} \) \((1 \leq i \leq Q)\) representing the determinate execution sequence for task \( i \). These genes are joined together to form a chromosome like \((r_{1}, r_{2}, r_{3}, \ldots r_{Q-1}, r_{Q})\).

4.2. Fitness function

In the problem of task scheduling, we should to find a sequence of tasks to get the minimum total outbound operation time (OOT). Thus we transfer the minimum objective into a maximum optimization problem to simplify the solution domain.

\[
f = 1/t_{total}
\]  
(26)

Where \( f \) is the transformed fitness value.

4.3. Initial population

For GA to evolve new candidate solutions, it needs an initial population. Considering this paper’s topic, we adopt the principle of “excellent genes first” to generate the initial population. Hence the retrieval task near the entrance of the aisle is regarded as excellent genes, and the excellent genes of each tier are regarded as excellent genes of each chromosome in the population.
4.4. Selection
The parent selection operator is an important process that directs a GA search. Two parents are selected from the solutions of a particular generation. There are many methods to select the parent generation. In this paper, the roulette wheel method is adopted, and the selection probability can be calculated as follows:

\[ p_k = \frac{f_k}{\sum_{k=1}^{M} f_k} \]  
(27)

Where \( f_k \) represents the fitness of chromosome \( k \).

4.5. Crossover
Progeny is produced by combining information from their parents’ chromosomes and have both the good parts that inherited from each parent's chromosome respectively. As each gene represents a particular retrieval task, there can not appear duplicated genes on chromosomes. Therefore we applied an improved single-point crossover. The location of the crossover point is randomly selected, and each of the two progenies inherits the genes before the location form the parents (based on principle of “excellent genes first”). As for the genes after the location, the two progenies only inherit those that are not duplicated, and then fill in the rest without repetition.

4.6. Mutation
After recombination, some progenies are mutated to provide a small amount of randomness and prevent the solution from falling into local optimum. The encoding and crossover methods determine the type of mutation. In this paper, we adopt the method of inverse mutation: The mutation operator first randomly selects two mutation position, then exchange these two genes.

4.7. Termination conditions
These steps are repeated until the maximum number of generations is reached, then we select the chromosome whose fitness value is the biggest as the optimum.

5. Experiment results and analysis
In this section, we do some experiments to illustrate the value of the proposed scheduling model and the solving algorithm, using MATLAB. The experimental information is collected from a real warehouse system, according to which the values of some basic device parameters are set as shown in table 2. In our experiment, the MSS consists of 10 aisles, 10 tiers and 50 storage columns. The total number of storage positions is 5000.

| Parameters | values | Parameters | values |
|------------|--------|------------|--------|
| \( X \)    | 50     | \( Q \)    | 40     |
| \( Y \)    | 10     | \( h \)    | 1\( m \) |
| \( Z \)    | 10     | \( a \)    | 0.6\( m \) |
| \( a_x \)  | 1.5\( ms^{-2} \) | \( t^e \)  | 2\( s \) |
| \( a_t \)  | 1.5\( ms^{-2} \) | \( t^f \)  | 2\( s \) |
| \( V_c \)  | 4\( ms^{-1} \)  | \( t^g \)  | 3\( s \) |
| \( V_l \)  | 3\( ms^{-1} \)  |           |        |

In the experiment, we focus on the order retrieval tasks in an individual aisle. The target locations of 40 retrieval tasks are generated based on a random storage policy as described in section 2, as shown in table 3.

| Task | column | tier | Task | column | tier | Task | column | tier | Task | column | tier |
|------|--------|------|------|--------|------|------|--------|------|------|--------|------|
| 1    | 3      | 4    | 11   | 16     | 5    | 21   | 32     | 3    | 31    | 8      | 6    |
| 2    | 4      | 8    | 12   | 19     | 7    | 22   | 34     | 9    | 32    | 17     | 10   |
Firstly we conduct an experiment on tier-captive MSS as control study. Since there is a shuttle at each tier, the MSS can get the minimum outbound operation time by the retrieval principle of “from near to far”. The handling sequence of these tasks obtained are as follows:

9→32→21→4→14→2→28→1→39→36→10→33→22→23→15→16→11→17→5→3→29→34→24→30→40→18→31→25→6→12→37→7→26→38→19→35→13→8→20→27

And the standard total outbound operation time is 489.69s.

Then we conduct the same experiment on tier-to-tier MSS, where only 5 shuttles are deployed to fulfill order retrieval tasks on these 10 tiers in an aisle. The crossover and mutation rates are usually set in the range of 0.4-0.99 and 0.001-0.1 respectively, in this paper we set them as 0.8 and 0.1 respectively. We get a set of solutions as indicated in table 4 after attaining the given maximum generations.

Table 4. solutions of tier-to-tier MSS calculated by GA.

| Population quantity M | Maximum generations N | LIT (s) | OOT (s) | Standard OOT (s) | Differential to standard total OOT (s) |
|-----------------------|-----------------------|--------|---------|-----------------|--------------------------------------|
| M=100 N=100           |                       | 323.65 | 681.34  |                 | 191.65                               |
| M=100 N=200           |                       | 289.87 | 647.57  |                 | 157.87                               |
| M=150 N=100           |                       | 290.74 | 648.44  |                 | 158.74                               |
| M=150 N=200           |                       | 233.68 | 591.37  |                 | 101.68                               |
| M=150 N=300           |                       | 207.13 | 564.82  |                 | 75.13                                |
| M=150 N=400           |                       | 207.07 | 564.73  |                 | 75.04                                |
| M=100 N=100           |                       | 323.65 | 681.34  |                 | 191.65                               |
| M=100 N=200           |                       | 289.87 | 647.57  |                 | 157.87                               |
| M=150 N=100           |                       | 290.74 | 648.44  |                 | 158.74                               |
| M=150 N=200           |                       | 233.68 | 591.37  |                 | 101.68                               |

Table 5. the determinate of two MSS based on tasks scale.

| Experimental Group | Q | LIT (s) | OOT (s) | Standard OOT (s) | Differentials (s) |
|-------------------|---|---------|---------|------------------|-------------------|
| 1                 | 40| 207.13  | 564.82  | 489.70           | 75.13             |
| 2                 | 60| 192.44  | 808.74  | 746.20           | 62.54             |
| 3                 | 80| 179.22  | 996.44  | 945.77           | 50.68             |
| 4                 | 100| 160.65 | 1215.06 | 1171.80          | 38.56             |
| 5                 | 120| 151.50 | 1406.30 | 1367.80          | 37.27             |
| 6                 | 150| 145.35 | 1748.37 | 1711.10          | 37.27             |
From the results in the table we can find that the minimum OOT (564.72s) is obtained by setting \( M=150, N=400 \) in the tier-to-tier MSS, which is higher than that in the tier-captive MSS.

Since the optimal solution can be calculated when population quantity and maximum generations are 150 and 400 respectively, we tested the disadvantage to standard total OOT of Tier-to-tier MSS under this setting with different amount of tasks, as shown in table 5 and figure 5.

The results in figure 5 show that the Tier-to-tier MSS has higher operation time than the Tier-captive MSS. However, we can find that the difference of operation time between the Tier-to-tier MSS and the Tier-captive MSS narrows with the number of tasks, which means the Tier-to-tier MSS has the similar efficiency to that of Tier-captive MSS with lower shuttles.

6. Conclusions

In this study, we formulate a travel time model based on the parallel pickup of a tier-to-tier shuttle and the serial operation of two kinds of lifts. We proposed an advanced genetic algorithm to find the optimal solutions. The experimental results and analysis indicated that the disadvantage of the Tier-to-tier MSS to tier-captive MSS narrows with the increase of the number of retrieval tasks. We found that the tier-to-tier MSS could achieve similar working efficiency with fewer shuttles in a relatively large task scale which lower system costs.

The genetic algorithm in this paper cannot figure out large-sized optimization problems within a required short time.

References

[1] K.J. Roodbergen, I.F.A. Vis A survey of literature on automated storage and retrieval systems European Journal of Operational Research, 194 (2) (2009), pp. 343-362.
[2] M. Fukunari. Analytical foundations for autonomous vehicle storage and retrieval systems using load transfer station based dwell point strategies Rensselaer Polytechnic Institute (2005)
[3] C.J. Malmborg. Conceptualizing tools for autonomous vehicle storage and retrieval systems International Journal of Production Research, 40 (8) (2002), pp. 1807-1822.
[4] P. Kuo, A. Krishnamurthy, C.J. Malmborg. 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 (10) (2007), pp. 2332-2346.
[5] P. Kuo, A. Krishnamurthy, C.J. Malmborg. Performance modelling of autonomous vehicle storage and retrieval systems using class-based storage policies International Journal of Computer Applications in Technology, 31 (3–4) (2008), pp. 238-248.
[6] L. Zhang, A. Krishnamurthy, C.J. Malmborg, Sunderesh S. Heragu. Variance-based approximations of transaction waiting times in autonomous vehicle storage and retrieval systems European
[7] G. Marchet, M. Melacini, S. Perotti, E. Tappia. Analytical model to estimate performances of autonomous vehicle storage and retrieval systems for product totes International Journal of Production Research, 50 (24) (2012), pp. 7134-7148.

[8] B.Y. Ekren. Performance evaluation of AVS/RS under various design scenarios: A case study International Journal of Advanced Manufacturing Technology, 55 (9–12) (2011), pp. 1253-1261.

[9] G. Marchet, M. Melacini, S. Perotti, E. Tappia. Development of a framework for the design of autonomous vehicle storage and retrieval systems International Journal of Production Research, 51 (14) (2013), pp. 4365-4387.

[10] S. Brezovnik, J. Gotlih, J. Bali, K. Gotlih, M. Brezonik. Optimization of an automated storage and retrieval systems by swarm intelligence Paper presented at the 2013 24th DAAAM international symposium on intelligent manufacturing and automation.

[11] P. Yang, L. Miao, Z. Xue, et al. An integrated optimization of location assignment and storage/retrieval scheduling in multi-shuttle automated storage/retrieval systems Journal of Intelligent Manufacturing

[12] T. Wauters, F. Villa, J. Christiaens, R. Alvarez-Valdes, G. Vanden Berghe A decomposition approach to dual shuttle automated storage and retrieval systems Computers and Industrial Engineering

[13] M. Jiang, P. Yang. Retrieval scheduling for end-of-aisle multi-shuttle automated storage and retrieval systems Paper presented at the 2017 4th international conference on industrial engineering and applications (ICIEA) at the Nagoya Institute of Technology

[14] Y. Wang, S. Mou, Y. Wu. Task scheduling for multi-tier shuttle warehousing systems International Journal of Production Research, 53 (19) (2015), pp. 5884-5895.

[15] Y. Wang, S. Mou, Y. Wu. Storage assignment optimization in a multi-tier shuttle warehousing system Chinese Journal of Mechanical Engineering (English Edition), 29 (2) (2016), pp. 421-429.

[16] W. Yang, X. Wang, T. Yue, J. Zhang, T. Wang. Slotting optimization of double-loading multi-shuttle automated storage and retrieval system Packaging engineering, 39 (7) (2018), pp. 173-179.

[17] Melis Küükyaar, Ekren B Y , Lerher T . Cost and performance comparison for tier-aptive and tier-to-tier SBS/RS warehouse configurations[J]. International Transactions in Operational Research, 2020(2).

[18] Wang Y , Liu Z , Huang K , et al. Model and solution approaches for retrieval operations in a multi-tier shuttle warehouse system[J]. Computers & Industrial Engineering, 2020, 141(Mar.):106283.1-106283.9.

[19] H. Mengtao and C. Xiao, "Logistics Vehicle Scheduling Based on Genetic Algorithm," 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 2019.