Joint Optimization of Item Location Assignment and Task Scheduling for Tier-to-Tier Shuttle-Based Storage and Retrieval System (SBS/RS)

Billy Sagara
Graduate School of Science and Technology
Sophia University
Tokyo, Japan
sagaraBilly@gmail.com

Takashi Irohara
Faculty of Science and Technology
Sophia University
Tokyo, Japan
irohara@sophia.ac.jp

Abstract

The application of the Automated Storage and Retrieval System (AS/RS) in the warehouse has been improved significantly over the last few decades. In the recent years, a new type of AS/RS, Shuttle Based Storage and Retrieval System (SBS/RS) has taken many industries interest by storm due to its flexibility and throughput performances. This paper studies the mathematical optimization of item location assignment and task scheduling in SBS/RS mainly with tier-to-tier system which allows the shuttle to transport between tiers using a lift system. Since this system is mostly applied in giant e-commerce industries, considering order data set and multiple stock keeping unit (SKU) in the mathematical model formulation will give a significant benefit in the real-life scenario. This paper also proposed a modified genetic algorithm to solved both item location assignment and task scheduling with a good computational time result for large size instances within a reasonable amount of time.

Keywords
Mathematical Optimization, Shuttle Based Storage and Retrieval System, Genetic Algorithm

1. Introduction
Today, manufacturing industries have grown fast into a new era (Industrial 4.0) which utilizes more sophisticated automation technology in their process including artificial intelligence (AI) and machine learning. By using the automation technology, companies can save more labor costs as well as providing a better floor space with better movements. In the case of warehouse and storage logistics system, AS/RS is one of the most commonly used among giant manufacturing industries. Not only can give the best throughput performance, but it is also flexible for many different types of items depending on the design of the system itself.

There are many different types of AS/RS that has already been found such as: unit-load AS/RS, deep-lane AS/RS, mini-load AS/RS, man-on-board AS/RS, vertical lift storage module (VLSM), etc. However, Shuttle Based Storage and Retrieval System (SBS/RS) is the newest AS/RS type and it is considered to be the best one in comparison with other types. According to the study literature conducted by Kosanic et al. (2018), it has been proven that the system performs better than mini-load AS/RS. Since the system consists of 2 sub-systems (lift and shuttle), it greatly increases the efficiency of the warehouse part-picking operation. Nowadays, the application of SBS/RS is increasing significantly especially in pharmacy and e-commerce industries such as Amazon, Johnson & Johnson, or Alibaba.

Based on the design, SBS/RS can be divided into two types: tier-captive and tier-to-tier. In tier-captive SBS/RS, there is only one shuttle dedicated for each tier in the rack and they are not able to travel to other tiers. On the contrary, tier-to-tier design allows the shuttle to travel between tiers in which total number of shuttles is usually less than the total number of the tiers itself. This allows tier-to-tier SBS/RS to have a better utilization rate compared with tier-captive design.

In the e-commerce industry, demand from the customer usually comes in a form of multiple items with multiples of SKU. In this case, SBS/RS should be able to perform all the tasks as efficient as possible since time is very important and directly proportional with customer satisfaction. Therefore, this paper formulates the mathematical model for the task scheduling and item location assignment with the objective of reaching the best time, as minimum as possible. This paper also considers a time penalty given if the task is done later than expected. Because it is usually done in sequence according to the first-come-first-served (FCFS) basis. For better computational time results, a modified genetic algorithm is also proposed in this paper. To summarize the purpose of this research, the objective of this research is listed as follows:

(1) Develop a mathematical formulation for optimization of SBS/RS system considering order data set and multiple SKU which represents real-life dataset.
(2) Develop an algorithm approach for faster time computation and better capabilities for large instances.
(3) Evaluate the result for both method (exact solution and algorithm solution).

2. Literature Review

SBS/RS is a still considered a new technology and there are not many published papers regarding the optimization using mathematical modeling approach. Most of the papers published in this system scope are using simulation modeling and analytical approximation. Since, SBS/RS shares the same mechanism with Automated Guided Vehicle (AGV) and other AS/RS in general, taking a bit of references from those systems would be beneficial. Survey literature conducted by Kosanic et al. (2018), gives a detailed summarization of all research regarding SBS/RS that has been done until 2018. According to that literature, almost all of the researches that has been conducted are using simulation and analytical approach. Survey literature of AS/RS research by Roodbergen et al. (2009), i.e. summarization of all AS/RS research until 2009, is also included in the reference to give details about the AS/RS system scope and control issues. Fukunari (2005) states that SBS/RS combines the flexibility of AGV vehicle with the high-speed accessibility of AS/RS which are popular in the Europe and United States.

In the SBS/RS research, Lerher et al. (2012), has conducted a simulation modelling for the tier-captive SBS/RS to investigate both lift and shuttle sub systems with cycle time and throughput performance evaluation. Based on the result, they recommend the tier-to-tier configurations in which allowing the system to use lesser number of shuttles than the rack tiers. Lerher et al. (2015), also presented an analytical travel time model for travel time computation of SBS/RS. Ha et al. (2018), conducted a simulation modelling for the tier-to-tier SBS/RS to investigate the best control strategies. The result proves that mathematical modeling optimization will play an important role for achieving the best solution. Ekren et al. (2018), has developed an analytical model-based tool which provides performance estimations by changing the input parameters such as: travel distance, shuttles and lift velocity, acceleration/deceleration, etc. Zhao et al. (2018), has analyzes tier-to-tier SBS/RS for shuttle and lift system coordination. This paper models the system as a semi-open queuing network (SOQN) that is solved by the approximate mean value analysis (AMVA) algorithm. Borovinsek et al. (2019), introducing a solution procedure for multi-objective optimization of SBS/RS. The objectives considered in the paper are the minimization of average cycle time, energy consumption, and investment cost. Li et al. (2022), conducted research focusing on the joint optimization of multi-order order batching for SBS/RS. The paper introduced a mathematical model with the objective to minimize the carbon emission level. Zou et al. (2016), have built an analytical model based on fork-join queuing network for the parallel movement of lift and shuttles in SBS/RS and tested it by simulation modeling.

In the AS/RS research, Yu and De Koster (2012) presented a new heuristic policy, i.e. Percentage Priority to Retrievals with Shortest Leg (PPR-SL) for multi-deep automated storage system. Yang et al. (2015), proposed an integer programming model for joint optimization of location assignment and storage/retrieval scheduling in multi-shuttle AS/RS. They also proposed a variable neighborhood search heuristic method for better time computation. Kazemi et al. (2019), propose a new heuristic method solving the same model assumptions using Ant Colony Optimization (ACO) algorithm for generating initial solutions and Adaptive Large Neighborhood Search for the general solutions searching steps. Jiang and Yang (2017) proposed an integer programming model for retrieval scheduling problem of end-of-aisle multi-shuttle AS/RS. Mirzaei et al. (2021), studies order picking and proposed a new storage policy i.e. integrated cluster allocation (ICA) to minimize time travel of retrieval operation in considering product turnover and affinity rate. Wauters et al. (2016), introducing a decomposition approach for optimizing both location assignment and task sequencing in dual shuttle cranes of mini-load AS/RS.

In the case of AGV, mathematical optimization for minimizing earliness and tardiness has been proposed by Fazlollahtabar et al. (2015), which considers a job shop flow process. The paper also introduces heuristics algorithm and evaluates it with the other past heuristic approaches. Hsueh (2010) introduced a new design of bi-directional AGV. This design allows two AGVs to exchange their loads and their location destination which means exchange demand tasks as well. Sensitivity analysis also conducted to investigate the influence the exchange. Polten and Emde (2021) study the scheduling of storing and retrieval operation of unit-load AGV system in a very narrow aisle. Two access policies are proposed along with the mixed integer programming (MIP) model.

Since SBS/RS relatively a new system, most of the past studies are mostly about the analytical model and simulation approaches because of its complexity of the mechanism. This paper proposes the mathematical modeling optimization approach with the consideration of order data set and multiple SKU. The model later is solved by the exact method also and by the proposed modified genetic algorithm especially for solving large instances problem.

3. System Description

SBS/RS mainly consists of two sub systems: lift/elevator system and the shuttle system. The shuttle system is the system that consists of multiple shuttles which can move horizontally (x-axis) through the rack column. While the lift system is the system that taking in charge of vertical movement (y-axis). For the tier-captive design, the lift system is intended to transport the item that is delivered by the shuttles. The shuttles are already been at the desired tier and each tier has one of them. As for the tier-to-tier design, the lift will transport the shuttle itself to the desired tier allowing the system to work with less shuttles. Fig.1 shows the detailed design of the tier-to-tier SBS/RS.

Both storing and retrieval operation will be considered in this model. Both of them do not have any differences in terms of time travel cost since the movement operation is similar. Although, it will affect differently for the status of the rack which will also affect the task scheduling (explained in Section 4) since item type (SKU) is considered. The time travel for each sub system will be added and input as a cost parameter in the objective function as well as the time for the lift to take/release the shuttle.

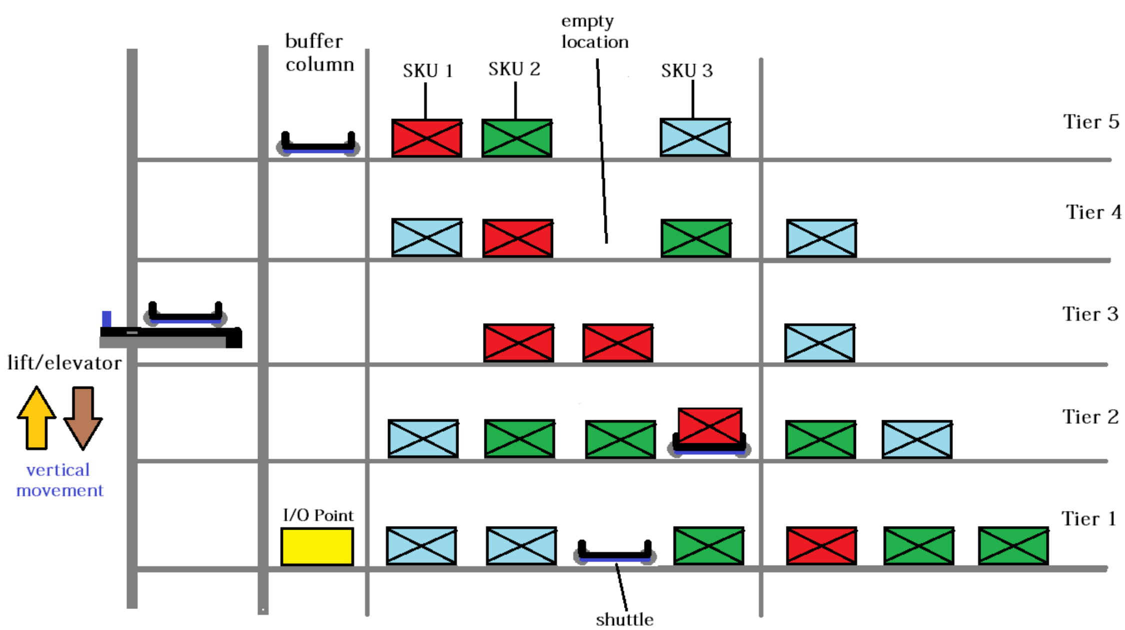

Figure 1. Tier-to-tier SBS/RS configuration

3.1 System Assumptions

In order to facilitate the research, the following assumptions are made for the system:

(1) A single rack aisle is considered in this research. The size of the rack is determined as a parameter.
(2) Multiple tiers are considered in the model, although only a single lift and a single shuttle are considered
(3) Each location in the rack has the same specification and separated with the same distance, same as each tier
(4) Stock replenishments are considered in the model by the storing operation.
(5) Operation step of the shuttle system follows the single command cycle (SCC).
(6) The motion parameters of with-load and without-load are the same for both lift and shuttle system.
(7) The maximum velocity of both lift and shuttle are known and determined and remain constant (unchanged)
(8) There is no tier 0 in the system. Tier 1 column 0 is considered the buffer point of the shuttle (I/O point).
(9) Time for the lift to take shuttle inside is the same with releasing the shuttle.

3.2 Travel Time Profile

This model uses time as its main objective cost for the minimization problem. Since acceleration and deceleration is considered, the movement can be divided into two types as shown in Fig.2. System parameter notation is described respectively as follows: $\boldsymbol{v_x}$= shuttle speed (m/s), $\boldsymbol{a_x}$= shuttle acceleration (m/$\boldsymbol{s^2}$), $\boldsymbol{d_x}$= distance each location/column (m), $\boldsymbol{v_y}$= lift/elevator speed (m/s), $\boldsymbol{a_y}$= lift/elevator acceleration (m/$\boldsymbol{s^2}$), $\boldsymbol{d_y}$= distance between each tier (m), and $\boldsymbol{r^t}$= time for the lift to take/release the shuttle.

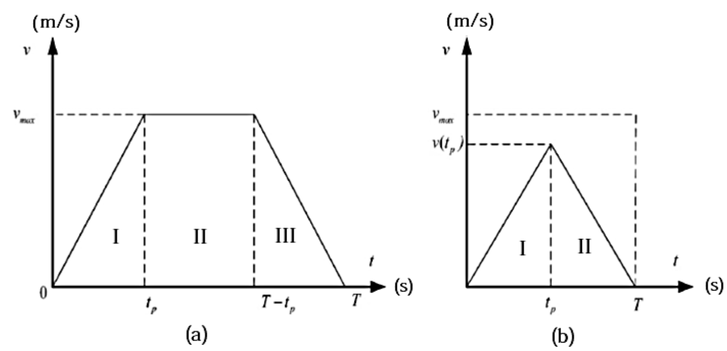

Figure 2. Velocity-Time Profile

(1) The subsystem has reached its maximum velocity before the deceleration (Fig.2a). In this case, time calculation is divided into three stages: time to accelerate from 0 to maximum velocity ($\boldsymbol{v^{max}}$), time for traveling at constant $\boldsymbol{v^{max}}$, and time for decelerating from $\boldsymbol{v^{max}}$ to 0.
(2) The subsystem has not reached its maximum velocity yet (Fig.2b). In this case, it only divided into two stages: time to accelerate from 0 to certain velocity ($\boldsymbol{v}$) and time to decelerate from that velocity ($\boldsymbol{v}$) to 0.

Based on the time profile above, the time cost calculation for each group can be formulated as follows:

- Time for the lift to move from tier 1 to tier j and go back to tier 1 ($\boldsymbol{t_j^l}$), is shown in the Eq.(1) below.

$$t_j^l = \begin{cases} 2 \times \left[\dfrac{2 \times v_y}{a_y} + \dfrac{(j-1) \times d_y - {v_y}^2/a_y}{v_y} \right] & (j-1) \times d_y > \dfrac{{v_y}^2}{a_y} \\ 2 \times \left[2 \times \sqrt{\dfrac{(j-1) \times d_y}{a_y}} \right] & (j-1) \times d_y \le \dfrac{{v_y}^2}{a_y} \end{cases} \tag{1}$$

- Time for the lift to move from tier 1 to tier j and go back to tier 1 ($\boldsymbol{t_i^s}$), is shown in the Eq.(1) below.

$$t_i^s = \begin{cases} 2 \times \left[\dfrac{2 \times v_x}{a_x} + \dfrac{i \times d_x - {v_x}^2/a_x}{v_x} \right] & i \times d_x > \dfrac{{v_x}^2}{a_x} \\ 2 \times \left[2 \times \sqrt{\dfrac{i \times d_x}{a_x}} \right] & i \times d_x \le \dfrac{{v_x}^2}{a_x} \end{cases} \tag{2}$$

- Total additional time for the lift to take/release shuttle ($\boldsymbol{t_j^r}$), is shown in the Eq.(3) below. Notice the value of $\boldsymbol{r^t}$ needs to be multiplied by 4 since the take/release operation happens four times.

$$t_j^r = \begin{cases} r^t \times 4 & j > 1 \\ 0 & j \leq 1 \end{cases} \tag{3}$$

Hence, the total time travel cost that is used in the model is formulated in the Eq.(4) below:

$$c_{i,j} = t_j^l + t_i^s + t_j^r \tag{4}$$

4. Problem Description

The model considers the order data set in which each order has several tasks (denoted as $\boldsymbol{k}$). All tasks that are listed in the order data set must be completed without any exceptions. As an example, consider an order data set for small instances of 15 tasks with $\boldsymbol{k = 3}$ that is shown in Table.1.

Table.1 Order Data Set

Demand Data Set			
Order	**task(g)**	**item type**	**operation**
1	1	A	storing
	2	B	storing
	3	A	storing
2	4	B	storing
	5	B	storing
	6	C	storing
3	7	B	retrieval
	8	B	retrieval
	9	A	retrieval
4	10	C	retrieval
	11	B	retrieval
	12	C	retrieval
5	13	C	storing
	14	C	storing
	15	A	storing

Each task in the order data set has an operation status whether it is storing or retrieval. Retrieval operation means demand from the customer while storing operation means demand item input from the supplier. The model considers the storage/rack status which the item availability in the rack will be taken into a consideration. All the tasks are assumed to be processed continuously since time is off the essence. The system is not allowed to do a retrieval task if the corresponding item is not available in the rack. For example, based on the data set given in Table.1, the system is not allowed to do task 4 (retrieval item B) if item B is not available in the rack at that moment. Since each task within one order comes from the same customer/supplier, it would be unpermitted to serve other customer demands while the corresponding customer demand still has not been finished. This means that we should not do other tasks in another order until we finished doing all the tasks in the corresponding order. To illustrate this, consider Table.1 as an example. Suggest that we need to do task 4 (storing item B) first. After task 4 is done, the system is not allowed to do other tasks before task 5 and task 6 are done, since they are in the same order (order 2). This task clustering assumption is one of the important assumptions that will bring benefit in real scenario.

4.1 Problem Assumptions

In order to facilitate the research, the following assumptions are made for the system:

(1) Multiple of tasks and multiples of SKU are considered.
(2) Each order contains the same number of tasks which denoted as $\boldsymbol{k}$.
(3) The rack system of each location in the rack can be placed with any SKU without any restriction.
(4) Each location can only be filled with one item.

(5) The systems are not allowed to do other tasks in other order before all the tasks in the corresponding order is finished (task clustering assumption).
(6) The systems are not allowed to retrieve the item if the corresponding SKU is not available in the rack.
(7) The initial storage/rack status ($\boldsymbol{s_{iju0}}$) is set to empty (0 item) to all location. Although non-empty initial rack status can be considered in the model since it can be defined as a parameter.

4.2 Time Penalty

Since normally we followed the FCFS rule for serving the customer, time penalty is considered in the model to cope such extreme task sequence solution. In the business point of view, the first customer usually will be served the first. Same goes with the second customer which usually will be served the second. Since time directly proportional with customer satisfaction, this model will add time penalty as one of the costs in the objective function. The time penalty is formulated in the Eq.(5) as follows:

$$p_{g,f} = \begin{cases} (f-g) \times t^p & f > g \\ 0 & f \leq g \end{cases} \tag{5}$$

The time penalty $\boldsymbol{p_{g,f}}$ represents the tardiness (lateness) cost for doing task g at the f iteration. The value of $\boldsymbol{t^p}$ is determined as a parameter that indicates how big the penalty will be given (in seconds) for each difference in the sequences. The model did not consider any penalty for the earliness ($\boldsymbol{p_{g,f} = 0}$).

5. Problem Formulation

The joint optimization integer programming model is proposed to formulate the problem. The goal is to minimize the total time value of the system to finish all the operation according to the demand data set. There are two main decision variables established in the model: location assignment ($\boldsymbol{x_{ijuf}}$) and task scheduling ($\boldsymbol{y_{guf}}$). $\boldsymbol{x_{ijuf}}$ determine which location that the shuttle is going to visit at iteration *f*. Meanwhile, $\boldsymbol{y_{guf}}$ will determine which task that the system will do at iteration *f*. Parameter notations are shown in the Table.2 and sets notation in Table.3. The objective function used in the model consists of two parts which are travel cost and penalty cost. Travel cost ($\boldsymbol{c_{i,j}}$) will occur depending on the item location assignments while penalty cost ($\boldsymbol{p_{g,f}}$) occur depending on task scheduling. Both factors might have a trade-off relationship, since the fastest travel time solution is not always has a similar sequence with FCFS rule. The formulation is as follows:

Table.2 Parameter Notation

$\boldsymbol{q}$	Total number of tasks
$\boldsymbol{n}$	Total number of locations at each tier
$\boldsymbol{m}$	Total number tiers
$\boldsymbol{k}$	Total number of tasks in one order
$\boldsymbol{v}$	Total number of SKU
$\boldsymbol{c_{ij}}$	Total cost for the shuttle to travel from I/O point to location i tier j
$\boldsymbol{p_{gf}}$	Penalty cost if task g is done at iteration f
$\boldsymbol{q_{gu}}$	Matrix demand data set of task g, item type u (= 1 if storing, = -1 if retrieval, and = 0 otherwise)

Table.3 Sets Notation

$\boldsymbol{N}$	Set of all locations at each tier
$\boldsymbol{M}$	Set of all tiers
$\boldsymbol{V}$	Set of all locations with I/O points ($N + \{0\}$)
$\boldsymbol{Q}$	Set of all tasks
$\boldsymbol{U}$	Set of all SKU
$\boldsymbol{S}$	Set of all tasks in the beginning of an order

Decision variables:

$x_{ijuf} = \begin{cases} 1 \text{ indicates the shuttle travels to location } i \text{ tier } j \text{ for SKU } u \text{ at the iteration } f \\ 0 \text{ otherwise} \end{cases}$

$y_{guf} = \begin{cases} 1 \text{ indicates task } g \text{ SKU } u \text{ is done at the iteration } f \\ 0 \text{ otherwise} \end{cases}$

$z_{ijguf} = \begin{cases} 1 \text{ indicates the shuttle is assigned to location } i \text{ tier } j \text{ for task } g \text{ SKU } u \text{ at the iteration } f \\ 0 \text{ otherwise} \end{cases}$

$s_{ijuf} = \begin{cases} 1 \text{ indicates there is an SKU } u \text{ on location } i \text{ tier } j \text{ iteration } f \\ 0 \text{ otherwise} \end{cases}$

$b_f =$ binary variable constraint linearization for iteration f

Objective function:

$$Min \sum_{i}^{n}\sum_{j}^{m}\sum_{u}^{v}\sum_{f}^{q} c_{ij}x_{ijuf} + \sum_{f}^{q}\sum_{g}^{q}\sum_{u}^{v} p_{gf}y_{guf} \tag{6}$$

Subject to:

$$\sum_{i}^{n}\sum_{j}^{m}\sum_{u}^{v} x_{ijuf} = 1, \qquad \forall f \in Q \tag{7}$$

$$\sum_{u}^{v}\sum_{f}^{q} y_{guf} = 1, \qquad \forall g \in Q \tag{8}$$

$$\sum_{u}^{v}\sum_{g}^{q} q_{gu}y_{guf} + 2b_f \geq 1, \qquad \forall f \in Q \tag{9}$$

$$\sum_{u}^{v}\sum_{g}^{q} q_{gu}y_{guf} + 2b_f \leq 1, \qquad \forall f \in Q \tag{10}$$

$$\sum_{u}^{v}\sum_{i}^{n}\sum_{j}^{m} s_{iju0} = 0 \tag{11}$$

$$\sum_{u}^{v} s_{ijuf} \leq 1, \qquad \forall i \in N, \forall j \in M, \forall f \in Q \tag{12}$$

$$s_{ijuf} - s_{ijuf-1} - q_{gu}\, z_{ijguf} \leq 1 - z_{ijguf}, \qquad \forall i \in N, \forall j \in M, \forall g \in Q, \forall f \in Q, \forall u \in U \tag{13}$$

$$s_{ijuf-1} + q_{gu}\, z_{ijguf} - sc_{ijuf} \leq 1 - z_{ijguf}, \qquad \forall i \in N, \forall j \in M, \forall g \in Q, \forall f \in Q, \forall u \in U \tag{14}$$

$$s_{ijuf} - s_{ijuf-1} \leq x_{ijuf}, \qquad \forall i \in N, \forall j \in M, \forall f \in Q, \forall u \in U \tag{15}$$

$$s_{ijuf-1} - s_{ijuf} \leq x_{ijuf}, \qquad \forall i \in N, \forall j \in M, \forall f \in Q, \forall u \in U \tag{16}$$

$$\sum_{i}^{n}\sum_{j}^{m} z_{ijguf} - 1 \leq 1 - y_{guf}, \qquad \forall g \in Q, \forall f \in Q, \forall u \in U \tag{17}$$

$$1 - \sum_{i}^{n}\sum_{j}^{m} z_{ijguf} \leq 1 - y_{guf}, \qquad \forall g \in Q, \forall f \in Q, \forall u \in U$$

$$z_{ijguf} \leq x_{ijuf}, \quad \forall i \in N, \forall j \in M, \forall g \in Q, \forall f \in Q, \forall u \in U \tag{18}$$

$$z_{ijguf} \leq y_{guf}, \quad \forall i \in N, \forall j \in M, \forall g \in Q, \forall f \in Q, \forall u \in U \tag{19}$$

$$z_{ijguf} \geq x_{ijuf} + y_{guf} - 1, \quad \forall i \in N, \forall j \in M, \forall g \in Q, \forall f \in Q, \forall u \in U \tag{20}$$

$$\sum_{u}^{v} \sum_{d=0}^{k-1} y_{g+d,u,f+d} - k \leq (1 - y_{guf})\, k, \quad \forall g \in S, \forall f \in \{1,2, \dots q-k+1\} \tag{21}$$

$$k - \sum_{u}^{v} \sum_{d=0}^{k-1} y_{g+d,u,f+d} \leq (1 - y_{guf})\, k, \quad \forall g \in S, \forall f \in \{1,2, \dots q-k+1\} \tag{22}$$

$$y_{guf} = 0, \ \forall g \in S, \forall u \in U, \forall f \in \{q-k+2, \dots, q\} \tag{23}$$

$$s_{ijuf}, x_{ijuf}, z_{ijguf}, y_{guf}, b_f, \in \{0,1\} \tag{24}$$

(25)

Constraint (7) denotes that shuttle can only visit one location to store or retrieve one item each iteration. (8) denotes that each task must be done once and only once. (9) and (10) denote that the system can only do one task in each iteration whether it is retrieval or storing operation. (11) denotes that storage/rack status are empty in the beginning (iteration 0). This constraint will be deleted if the initial status is not empty. In this case, $\boldsymbol{s_{iju0}}$ will become a data set (parameter) which can be adjusted according to rack initial status. (12) denotes the maximum capacity of each location in the rack equals to one item. Constraint (13) until (18) talks about the storage/rack status update rule. (13) and (14) denotes that if the shuttle does a storing/retrieval operation, the status in that corresponding location will be updated (+1 if storing and -1 if retrieval). (15) and (16) denote that all other location status that the shuttle did not visit, will be the same as the status at the iteration before. (17) and (18) denote the equality constraint of $\boldsymbol{z_{ijguf}}$. Constraint (19) until (21) denote the relationship between decision variable $\boldsymbol{z_{ijguf}}$ and decision variables $\boldsymbol{x_{ijuf}}$ and $\boldsymbol{y_{guf}}$. (22) and (23) denote that the system is not allowed to do other task in other order before finishing all the task in the corresponding order (task cluster constraint). (24) denotes that beginning task in an order cannot be placed at the several last iterations depending on the value of *k*. (25) denotes all decision variable is binary.

6. Proposed Algorithm

Since the problem is about task scheduling and item location assignment problem, like any other related field research paper, this problem is considered as a NP-Hard problem. Since there are two main decision variables in the model, the algorithm is divided into two different approaches: item location assignment and task scheduling. For the item location assignment, the algorithm follows the approach of priority-based dispatching rule. As for the task scheduling, genetic algorithm approach is used. Since it combines both genetic algorithm and priority-based dispatching rule, hence the name becomes modified GA.

6.1 Priority-based Dispatching Rule

In this case, the priority is set based on the total number of retrieval task in the corresponding SKU. For example: if the total number of retrieval tasks for item type a, b, c respectively are 3, 7, 2, then item b will be prioritized first to be stored on the shortest distance location followed by item a and item c. If there are two or more items have the same total amount of retrieval tasks, then they have the same probability to be prioritized first. The probability value ($\boldsymbol{w_{ul}}$) for SKU u to be assigned in the *l*-th shortest distance location is calculated in the Eq.(26) as follows:

$$w_{ul} = \frac{s_u + l - 1}{max(s)} \times 100\% \tag{26}$$

Variable $\boldsymbol{s_u}$ is denoted as the total number of retrieval task for SKU *u*, while max(s) means the highest value (modus) of s for all SKU. From the Eq.26 above, the probability value of $\boldsymbol{w_{ul}}$ will increases for each *l*-th shortest distance location. This means the probability $\boldsymbol{w_{ul}}$ will get bigger as the *l* is getting bigger. The algorithm will always calculate the first shortest distance location first before moving on to the next shortest distance.

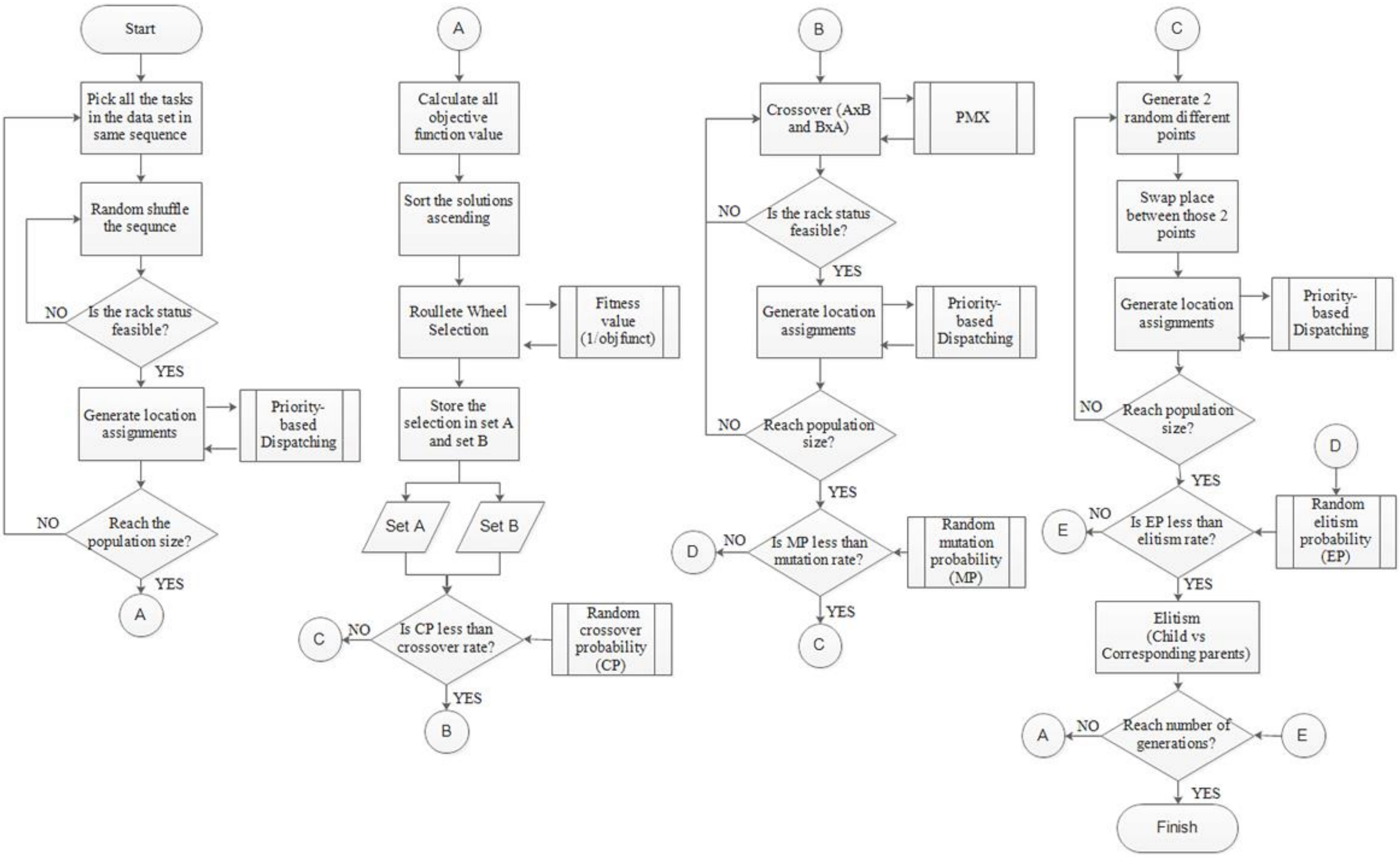

Figure.3 Modified Genetic Algorithm Flowchart

6.2 Genetic Algorithm

Like any other genetic algorithm (GA) approach, in this case, the steps are divided into 5 major sections: random generation, fitness calculation and crossover selection, crossover, mutation, and last selection (elitism). Since task scheduling in this problem is about generating the best sequence, genetic algorithm is used and it has already been proven its capabilities by researchers. The flowchart concept for all GA steps is shown in Fig.3.

In random generation step, as a first step, the algorithm picks all the orders in the data set as it is. Then, the algorithm will randomly shuffle the sequence using a built-in python random shuffle which follows the Fisher-Yates random shuffle method. The shuffle needs to be per order since it is bound to the task clustering constraint (see Section 4 for task clustering assumption explanation). In this algorithm, the fitness calculation is ranked based on the single objective function value. The algorithm will automatically calculate every solution's objective function value, and sort it ascending. This means, the best solution will be at the top (first index) and the worst solution will be at the bottom (last index). After the solution is sorted, the selection process for the crossover method will begin by dividing the population solutions into two same size set (set A and set B). The dividing process is determined by the proportionate probability of roulette wheel selection which depends on the fitness calculation value. The fitness calculation for solution *i* is calculated in Eq.(27) below:

$$fitness_i = \frac{1}{objective\ function\ value_i} \times 1000 \tag{27}$$

The crossover operations for this task scheduling problem follows the Partially Mapped Crossover (PMX) method, since this method is commonly used especially for travelling salesmen problem (TSP). The cut points of the method are determined randomly where every value has the same probability (discrete uniform distribution with the range from 1 until total number of orders -1). As for the mutation, a single random swapped is used with both points also determined randomly. Lastly, another selection method is conducted at the end which is elitism. In this genetic algorithm, elitism is a step to compare the child solution with the corresponding parent solution before crossover.

7. Experimental Results

The experiment results is divided into three based on the size of the problem. Small size problem is the problem that takes no more than 1 hour for the exact method to find the best solution. Medium size problem is when the exact method cannot find the best solution within 1 hour but feasible solutions is found. Meanwhile, large size problem means the exact solution method cannot find a single feasible solution within 1 hour. All experiments (small, medium, and large) are conducted using the same system parameter respectively: $\boldsymbol{v_x = 2\, m/s}$, $\boldsymbol{a_x = 1\, m/s^2}$, $\boldsymbol{d_x = 1\, m}$, $\boldsymbol{v_y = 1\, m/s}$, $\boldsymbol{a_y = 1\, m/s^2}$, $\boldsymbol{d_y = 0.6\, m}$ with $\boldsymbol{r^t}$= 0.4 seconds. The size of the rack implemented is 3x10 (n=10 and m=3). Experiments parameter each trial will be explained in each section. Modified genetic algorithm parameter P x G respectively are notated as population size multiplied by number of generations. All experiments are using Gurobi 9.0.3 solver for the exact solution method. All the experiments are also conducted in a computer with specification of Intel Core-i7-7500U, 3.5GHz with 8GB RAM.

The result for small instances is shown in Table.4. Crossover rate, mutation rate, and elitism rate that are used respectively are: 90%, 1%, 2%. In this case, in terms of time, the algorithm shows a better performance compare with the Gurobi results. The percentage gap $\boldsymbol{G_e}$ are obtain information from Gurobi, while $\boldsymbol{G_m}$ is calculated in the Eq.(28) as follows:

$$G_m = \frac{Z_m - Z_e}{Z_e} \times 100\% \tag{28}$$

Table.4 Result for Small Size Instances

Experiment Case					Gurobi 9.0.3 (Exact Method)			Modified GA			
Instances	k	SKU (v)	tasks (q)	tp	Z_e	T_e (s)	G_e (%)	Z_m	T_m (s)	G_m (%)	P x G
1	2	2	10	1	61.556	5.82	0	61.556	4.97	0	50 x 20
2	3	3	12	1	93.567	13.08	0	93.567	2.08	0	20 x 50
3	3	3	15	1	120.110	76.83	0	120.110	24.78	0	50 x 50
4	3	3	15	1	93.110	33.92	0	93.110	24.49	0	50 x 50
5	3	3	15	1	110.039	37.47	0	110.039	24.45	0	50 x 50
6	2	3	20	0.3	110.711	326.30	0	110.711	252.94	0	200 x 50
7	4	3	20	0.3	184.737	208.56	0	184.737	119.07	0	200 x 50
8	4	4	20	0.5	160.737	222.81	0	160.737	212.49	0	100 x 150
9	3	3	21	1	184.394	493.61	0	184.394	264.22	0	200 x 50
10	5	3	25	1	258.791	623.69	0	258.791	353.44	0	200 x 50
11	5	4	25	1	233.791	921.72	0	233.791	612.27	0	200 x 100
12	3	3	30	1	236.949	1792.26	0	236.949	1229.11	0	200 x 150

There is no gap value between the Modified genetic algorithm with the Gurobi solution. The algorithm also performs better in terms of computational time ($\boldsymbol{T_m}$ compared with $\boldsymbol{T_e}$). Instance 1 and 2 shows the result for a relatively small task (solved under 15 seconds). Instance 3 until instance 5 are for testing the algorithm consistency for the same parameter k, v, q. Instance 6 and 7 is to show the difference if value k is different under the same circumstances. The result shows that higher number of k makes the objective function higher and time computation faster. Instance 6 and 8 is to show the result for different time penalty ($\boldsymbol{t^p}$) value, and higher k, v value. Instance 9 and 12 shows number of tasks affecting the objective function value greatly. Instance 10 and 11 shows higher number of SKU reducing the objective function value.

The result for medium size instances is shown in Table.5. Crossover rate, mutation rate, and elitism rate that are used respectively are: 85%, 1%, 2%. Population size and number of generations is set to 200. For giving a better comparison, percentage difference between the algorithm solution is notated as $\boldsymbol{Z_m/Z_e}$. From the result, there are some instances where the algorithm produce**s** the same result as Gurobi like shown in the instance 13, 15, and 16. The time for Gurobi to solve the problem is limited to 1 hour. The result of the algorithm shows better at trial 14, 17 and 18 not only the objective function, but also the time computation result.

Table.5 Result for Medium Size Instances

Experiment Case					Gurobi 9.0.3 (Exact Method)			Modified GA			
Instances	k	SKU (v)	tasks (q)	tp	Z_e	T_e (s)	G_e (%)	Z_m	T_m (s)	Z_m / Z_e (%)	P x G
13	3	3	30	1	262.877	3600	8.96	262.877	2567.25	100	200 x 200
14	3	4	30	1	277.960	3600	12.6	274.919	2653.35	98.9059577	200 x 200
15	4	3	32	1	295.244	3600	1.14	295.244	1880.87	100	200 x 200
16	5	4	35	1	394.754	3600	1.39	394.754	2946.19	100	200 x 200
17	5	4	35	1	328.286	3600	6.72	326.316	2912.07	99.3999135	200 x 200
18	4	3	36	1	303.919	3600	7.31	303.545	3271.44	99.8769409	200 x 200

The result for large size instances is shown in Table.6. Crossover rate, mutation rate, and elitism rate that are used respectively are: 90%, 1%, 2%. The population size used in these experiments are 250. Any greater population size (above 250) will increase the time for each generation very significantly. It is very noticeable that number generation for each trial is decrease since the parameter are getting bigger. All five algorithm experiments are terminated under 1 hour. Any experiments attempt to passed the 70 tasks, in most of the cases, ended up with memory limitation error (N/A). The solutions that are generated by the algorithm in this large size instances are compared with random generated solution. The value of random solutions ($\boldsymbol{Z_r}$) are the best objective function value among all random solutions that is generated within one hour. The percentage difference, notated as $\boldsymbol{P_m}$, is calculated in the Eq.(29) as follows:

$$P_m = \frac{Z_r - Z_m}{Z_r} \times 100\% \tag{29}$$

Table.6 Result for Large Size Instances

Experiment Case					Random solutions Z_r	Modified GA			
Instances	k	SKU (v)	tasks (q)	tp		Z_m	T_m (s)	P_m	P x G
19	4	3	40	1	408.003	355.746	3600	12.807994	250 x 221
20	4	4	40	1	508.223	447.423	3600	11.963252	250 x 215
21	5	3	50	1	523.149	492.796	3600	5.8019799	250 x 189
22	5	4	50	1	724.309	532.184	3600	26.525281	250 x 170
23	5	3	60	1	701.076	554.996	3600	20.836543	250 x 123

8. Conclusions

In this paper, mathematical optimization of item location assignment and task scheduling for tier-to-tier SBS/RS is studied for minimizing the total time. Many originality considerations are proposed such as the task clustering assumption and rack status update. As for cost, not only travel time cost that is considered, but also the penalty time due to the lateness of task operation. The author believes that the proposed mathematical model could benefit the SBS/RS research in the future. Furthermore, this paper also introduced an algorithm that capable of showing good results in comparison to the proposed mathematical model. Some important points regarding this proposed model that can be improved in the future are:

- Multiple of shuttles and multiples of lifting system
- Each order might contain a different number of task (multiple $\boldsymbol{k}$)
- Dual command cycle system for both shuttle and lift system

References

Borovinšek, M., Ekren, B. Y., Burinskienė, A., & Lerher, T., Multi-objective optimization model of shuttle-based storage and retrieval system, *Transport*, vol.32, no.2, pp.120-137,2017.

Ekren, B. Y., Akpunar, A., Sari, Z., & Lerher, T., A tool for time, variance and energy related performance estimations in a shuttle-based storage and retrieval system, *Applied Mathematical Modelling*, vol.63, pp.109-127,2018.

Fazlollahtabar, H., Saidi-Mehrabad, M., & Balakrishnan, J., Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms, *Robotics and Autonomous Systems*, vol.72, pp.131-138,2015.

Fukunari, Miki., *Analytical foundations for autonomous vehicle storage and retrieval systems using load transfer station based dwell point strategies*. Rensselaer Polytechnic Institute, 2005.

Ha, Y., and Chae, J., Free balancing for a shuttle-based storage and retrieval system, *Simulation Modelling Practice and Theory*, vol.82, pp.12-31,2018.

Hsueh, C.F., A simulation study of a bi-directional load-exchangeable automated guided vehicle system, *Computers & Industrial Engineering*, vol.58, no.4, pp.594-601,2010.

Jiang, M., and Yang, P., Retrieval scheduling for end-of-aisle multi-shuttle automated storage and retrieval systems, *4th International Conference on Industrial Engineering and Applications* (ICIEA), pp.300-304, IEEE, 2017.

Kazemi, M., Asef-vaziri, A., & Shojaei, T., Concurrent optimization of shared location assignment and storage/retrieval scheduling in multi-shuttle automated storage and retrieval systems, *IFAC-PapersOnLine*, vol.52, no.13, pp.2531-2536,2019.

Kosanić, N.Ž., Goran Z., Milojević, & Zrnić, N.D., A survey of literature on shuttle based storage and retrieval systems. *FME Transactions*, vol.46, no.3, pp.400-409,2018.

Lerher, T., Ekren, B. Y., Dukic, G., & Rosi, B., Travel time model for shuttle-based storage and retrieval systems, *The International Journal of Advanced Manufacturing Technology*, vol.78, no.9, pp.1705-1725,2015.

Lerher, T., Rosi, B., Ekren, Y., & Sari, Z., A model for throughput performance calculations of shuttle based storage and retrieval systems, *Technicki vjesnik*, vol.19, no.4, pp.709-715,2012.

Li, H., Lyu, J., Zhen, L., & Zhuge, D., A joint optimization of multi-item order batching and retrieving problem for low-carbon shuttle-based storage and retrieval system, *Cleaner Logistics and Supply Chain*, vol.4, p.100042,2022.

Mirzaei, M., Zaerpour, N., & De Koster, R., The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance, *Transportation Research Part E: Logistics and Transportation Review*, vol.146 p.102207,2021.

Polten, L., and Emde, S., Scheduling automated guided vehicles in very narrow aisle warehouses, *Omega*, vol.99, p.102204,2021.

Roodbergen, K.J., and Vis, I.F., A survey of literature on automated storage and retrieval systems, *European journal of operational research*, vol.194, no.2, pp.343-362,2009.

Wauters, T., Villa, F., Christiaens, J., Alvarez-Valdes, R., & Berghe, G. V., A decomposition approach to dual shuttle automated storage and retrieval systems, *Computers & Industrial Engineering*, vol.101,pp.325-337,2016

Yang, P., Miao, L., Xue, Z., & Ye, B., Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems, *Transportation Research Part E: Logistics and Transportation Review*, vol.79, pp.164-177,2015.

Yu, Y. and De Koster, R.B., Sequencing heuristics for storing and retrieving unit loads in 3D compact automated warehousing systems, *IIE Transactions*, vol.44, no.2, pp.69-87,2012.

Zhao, X., Zhang, R., Zhang, N., Wang, Y., Jin, M., & Mou, S., Analysis of the shuttle-based storage and retrieval system, *IEEE Access*, vol.8, pp.146154-146165,2020.

Zou, B., Xu, X., & De Koster, R., Modeling parallel movement of lifts and vehicles in tier-captive vehicle-based warehousing systems, *European Journal of Operational Research*, vol.254, no.1, pp.51-67,2016.

Biographies

Billy Sagara is a master's student at Science and Technology Department of Sophia University, Tokyo, Japan. He is scheduled to graduate from the university in September 2022 and earned his M.E. title. He obtained his bachelor (Industrial Engineering) in 2019 from Catholic Parahyangan University (UNPAR), Bandung, Indonesia. During the time when he was in bachelor, his bachelor thesis in which addressing the development of multi-input AS/RS design, was submitted to the IOP Conference Series of Material Science and Engineering.

Takashi Irohara received the B.E., M.E., and Doctor of Engineering degrees from Waseda University, Japan, in 1993, 1995, and 1998, respectively. Since 2010, he has been working as a professor at the Department of Information and Communication Sciences, Faculty of Science and Technology, Sophia University, Japan. He has published over 80 reviewed journal papers in the area of facility logistics (order picking, inbound/outbound truck scheduling in the warehouse, facility layout problem, material handling), supply chain management (inventory control, transportation, and vehicle routing problem), production scheduling and humanitarian relief logistics. He served as a board member of the International Foundation of Production Research (IFPR), APIEMS (Asia Pacific Industrial Engineering and Management Systems), Japan Industrial Management Association (JIMA), Japanese Material Handling Society (JMHS) and so on.