International Journal of Industrial Engineering, 30(2), 522-535, 2023

PERFORMANCE IMPACT OF DISPATCHING AND ROUTING IN AN AUTOMATED GUIDED VEHICLE SYSTEM

Inhye Bang[1], Byung-In Kim[1,*], and Yonggu Kim[2]

[1]Department of Industrial and Management Engineering
Pohang University of Science and Technology (POSTECH)
Pohang, Korea
[*]Corresponding author's e-mail: bkim@postech.ac.kr

[2]Module Equipment Development Team, Mechatronics Technology Research Center,
Samsung Display Co.
Yongin, Korea

Dispatching and routing are fundamental operational decisions in automated material-handling systems. Numerous studies have been conducted on these two operational decisions, with more focus being recently made on intelligent routing decisions. However, comparative studies between the effects of dispatching and routing methods have not been reported so far. In this study, we have investigated three dispatching and three routing algorithms and measured their impacts using a simulation model for an automated guided vehicle (AGV) system designed for a real-world production line, in which a grid-type material flow layout is used, and the AGVs need to stop before changing their direction of movement. Two routing algorithms are developed in this study. Simulation experiments revealed that both dispatching and routing algorithms affect the performance of the AGV system, although dispatching methods showed a more significant impact. Good dispatching and routing algorithms are mandatory to improve the overall performance of AGV systems.

Keywords: Automated Guided Vehicle; Dispatching; Routing; A* Algorithm; Turn Penalty.

(Received on October 19, 2022; Accepted on March 1, 2023)

1. INTRODUCTION

Automated material handling systems (AMHS) have been widely used in many areas and are recognized as important components of logistics systems (Chawla *et al*., 2019). AMHS makes use of different types of vehicles, such as automated guided vehicles (AGV), which move along marked lines. Additional examples include rail-guided vehicles that move along a straight fixed path, overhead host transport (OHT) that moves along a rail attached to the ceiling, and continuous flow transport that uses a conveyor (Nazzal and Bonder, 1998). In this study, we have treated the case of AGVs, although the findings are seemingly applicable to other AMHS.

Design and operational control issues are the two main problems in operating an AGV system efficiently. Design issues include guide path design, positioning of loading/unloading points and charging stations, and determination of the appropriate number of vehicles. Operational control issues include dispatching rules, path planning (routing), traffic management, and idle AGV positioning (Ganesharajah *et al*., 1998). In this study, we focus on dispatching and routing problems, which are fundamental operational decisions in many AMHS, as illustrated in Figure 1. When a load delivery request occurs in an AGV system, it is assigned by a dispatching rule or algorithm. The assigned AGV then moves to the loading point by choosing a path (or route) determined by a routing rule or algorithm. Arriving at the loading point, the AGV takes the load and moves to the unloading point destination determined by the same routing algorithm.

Owing to their fundamental importance, many studies have been conducted thus far on the investigation of dispatching and routing. With the novel advances in artificial intelligence, more interest is being drawn on intelligent routing decisions. However, studies comparing the effects of dispatching and routing methods are still lacking. In this study, we investigated three dispatching and three routing algorithms and measured their impacts using a simulation model for an AGV system. The three dispatching algorithms are the closest vehicle selection rule, reassignment-based dispatching rule (RBD), and Hungarian algorithm-based OHT reassignment (HABOR). The last two algorithms were proposed by the second author of this paper (Kim *et al*., 2007; Kim *et al*., 2009). The three routing algorithms tested here are the shortest route selection rule, dynamic and advanced dynamic A* algorithms. The A* algorithm is a well-known shortest-path search algorithm for one-to-one paths. In the dynamic A* algorithm, the route of each AGV is dynamically changed based on the A* algorithm using estimated travel

DOI: 10.23055/ijietap.2023.30.2.8657 **ISSN 1943-670X**

times between nodes, which are updated by the movements of the AGVs and exponential smoothing. However, the algorithm had some weaknesses when the travel times were not updated properly. Those weaknesses are described in Section 4.2.2. Thus, we developed a third routing algorithm in which the estimated travel times were refreshed periodically.

We conducted simulation experiments using AutoMod on a production line, in which AGVs must stop and turn when they change their moving direction. Even though simulations revealed that both dispatching and routing algorithms affect the performance of the AGV system, dispatching methods were found to have a more significant impact. In fact, the application of the dispatching rule results in a more effective reduction in the total lead time compared to the case of routing algorithms.

The remainder of this paper is organized as follows. In Section 2, a literature review on vehicle dispatching and routing in AGV systems is presented. The simulation environment is described in Section 3. In Section 4, three dispatching and routing algorithms are presented. The simulation results are presented in Section 5. Finally, the conclusions are presented in Section 6.

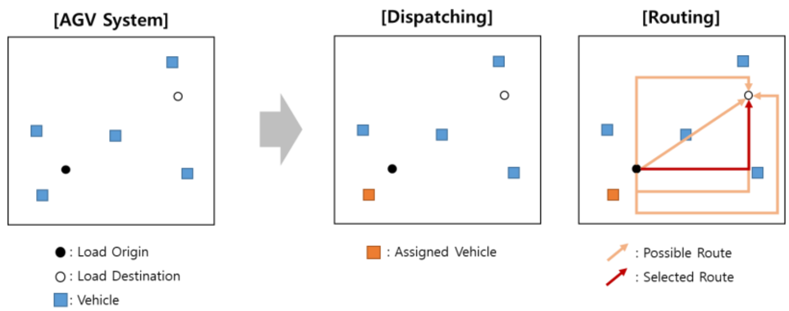

Figure 1. Dispatching and Routing Problems in AGV

2. LITERATURE REVIEW

Routing and dispatching are key issues directly related to the productivity of AMHS. Numerous studies have been conducted so far on the two operational decisions, with a more recent focus on intelligent routing decisions. However, to our knowledge, very few studies have applied good dispatching and routing algorithms together, and no study has compared their impacts. In Table 1, we present a summarized overview of previously reported relevant studies investigating routing and dispatching problems. Cited works were identified based on the subject they addressed, such as the dispatching problem, routing problem, dynamically changed routes, considered grid-type layout, guided path line, and vehicle turn time. In the last column, "Vehicle turn time" refers to the situations where the vehicle must stop before changing its direction. Thus, it needs deceleration before and acceleration after turning, and its turn time must be considered explicitly. Some studies have considered the turning time when the shortest time paths are searched.

Various routing algorithms have been proposed in the literature. Tuncer and Yildirim (2012) proposed a genetic algorithm (GA). Zhao and Fu (2012) and Dai *et al.* (2019) proposed ant colony optimization algorithms, and Tang, Zhu and Luo (2016) proposed particle swarm optimization. Recently, artificial intelligence models, including deep reinforcement learning (Raajan *et al.*, 2020) and Q(λ) learning (Hwang and Jang, 2020), have been proposed.

Bartlett *et al.* (2014) proposed a congestion-aware dynamic-routing algorithm. To consider congestion in an AMHS, their algorithm set edge weights representing the estimated travel time and rerouted vehicles based on these updated edge weights. When a vehicle completes its movement over the corresponding path segment, the edge weight is updated using exponential smoothing. Their routing algorithm outperformed a distance-based static routing algorithm. Lee *et al.* (2018) developed a similar shortest-time routing method that uses edge weights to reflect experienced congestion. The weight value of an edge is updated with the average speed of the vehicles, and the edge weight value is calculated by a distance-based cost multiplied by a congestion-based penalty factor.

The production line motivated by this study has a grid-type layout and requires the AGVs to stop before changing their direction of movement. From the literature, we found two studies that explicitly considered the vehicle turn time and reflected it in the routing decision. Fransen *et al.* (2020) applied turn penalties to grid-based AGV. Fransen and Eekelen (2021) considered a grid layout with turning costs and proposed an improved A* algorithm.

Although dispatching is very important and has a high potential for improving the productivity of an AMHS, recent studies on dispatching are rare. Ganesharajah, Hall, and Sriskandarajah (1998) summarized various dispatching rules. Kim *et al*. (2007) proposed a single reassignment at a time algorithm, RBD, and Kim *et al*. (2009) proposed a multiple reassignment at a time algorithm, HABOR. In this study, we implemented these two algorithms, the details of which are described in Section 4.

Table 1. Overview of Algorithms Reported in the Literature with Associated Characteristics.

Reference	Dispatching	Routing	Algorithm	Dynamic routing	Grid type	Guided line	Vehicle turn time
Bae and Chung, 2019	×	o	Primal-dual heuristic	×	o	o	×
Bartlett *et al*., 2014	×	o	Congestion-aware dynamic routing	o	×	o	× (1)
Corréa *et al*., 2007	o	o	Hybrid CP/MIP approach	o	×	o	×
Dai *et al*., 2019	×	o	Ant colony optimization with A* algorithm	×	o	×	×
Desaulniers *et al*., 2003	o	o	Column Generation	o	o	o	×
Fransen and van Eekelen, 2021	×	o	A* algorithm	×	o	×	o
Fransen *et al*., 2020	×	o	A* algorithm	o	o	×	o
Hwang and Jang, 2020	×	o	Q(λ) Learning	o	×	o	× (1)
Ki, Na, and Kim, 2019.	×	o	Route control with travel time prediction and random route selection	o	×	o	×
Kim *et al*., 2007	o	×	RBD	×	×	o	× (1)
Kim *et al*., 2009	o	×	HABOR	×	×	o	× (1)
Lee, Lee, and Na, 2018	×	o	Routing with congestion monitoring	o	×	o	× (1)
Martins *et al*., 2022	×	o	A* algorithm	×	o	×	o(2)
Mu *et al*., 2020	×	o	A* algorithm	o	o	o	o(3)
Nishi, Hiranaka, and Grossmann, 2011	×	o	Bilevel decomposition algorithm	o	×	×	×
Raajan *et al*., 2020	×	o	Deep reinforcement learning	o	×	×	×
Tang, Zhu, and Luo, 2016.	×	o	Particle swarm optimization	×	o	×	×
Tuncer and Yildirim, 2012	×	o	Genetic algorithm	o	o	×	×
Yuan *et al*., 2020	×	o	A* algorithm	o	o	o	×
Zhao and Fu, 2012	×	o	Ant colony optimization	×	o	×	×
This paper	o	o	RBD, HABOR, Advanced dynamic A* algorithm	o	o	o	o

(1) Dealing with OHTs on curved rails
(2) Considering the smoothness of the total path
(3) Considering the number of turns on the total path

As shown in Table 1, few studies have covered the dispatching and routing problems concurrently. Desaulniers *et al*. (2003) and Corréa *et al*. (2007) are the only papers that have considered the dispatching and routing problems concurrently. Desaulniers *et al*. (2003) proposed a mathematical formulation to determine the assignment of vehicle requests and a conflict-free routing solution to minimize the production delay cost. To solve the model, a column generation method and heuristic method were proposed; however, they were applied to small instances with no more than five vehicles. In addition, they assumed that the load occurrence was precisely scheduled, which hinders the application of their approach to dynamically changing factory environments with a large number of vehicles. Corréa *et al*. (2007) proposed a hybrid decomposition method using constraint programming (CP) and mixed integer programming (MIP). The decomposition framework consists of a master problem (scheduling problem modelled with CP) and a subproblem (conflict-free routing problem modelled with the MIP model with the time-space network). Their proposed method can solve small instances with up to six AGVs. Although the two abovementioned papers dealt with dispatching and routing problems concurrently, they did not analyze the effects of each and the simultaneous effects of the two.

The major contributions of our study can be summarized as follows.

1. We developed an advanced dynamic A* routing algorithm for a real-world production line, in which a grid-type material flow layout that requires AGVs to stop before changing their direction of movement is used by introduci ng an upper bound limit on the estimated link time and a refresh method. In the simulation experiments conducte d in this study, this algorithm showed improved productivity.
2. We simultaneously applied dispatching and routing algorithms by testing three dispatching and three routing algo

rithms and measured their impact. Our study is the first to measure the respective impacts quantitatively.

3. We demonstrated that dispatching methods have more significant impacts compared to routing methods. For a sa mple production line, the HABOR dispatching algorithm could reduce the lead time by 28.8 %, whereas the adva nced dynamic A* routing algorithm could reduce the lead time by only 4.5 %.
4. The impact of the simultaneous application of dispatching and routing algorithms was significant. The applicatio n of HABOR and Advanced Dynamic A* resulted in a 30.4 % reduction in the average total lead-time.

3. SIMULATION ENVIRONMENT

For our simulation, we used an AGV system, which covers part of a real production factory of Samsung Display, a global display manufacturer. The environment setup was similar to that of the actual process at the sample factory. The logistics process is to move display panels for production. When the display panels exit the machines, trays are used to carry them, and they can then be loaded onto AGVs. Therefore, there are two types of loads: panel and tray.

Figure 2 shows the simulated partial production line. The area of the production line is 12 m × 90.5 m, with 20, 24, 12, and 8 type machines. Each machine of types A, B, C, and D has a different configuration for the input and output ports, i.e., one IN and one OUT (separately), one IN, one OUT, and one OUT port, respectively. If a machine has an IN port, it can receive a load, whereas if it has an OUT port, it can generate a load delivery request. Note that some machines have IN ports only, and some have OUT ports only because we set the boundary of the target AGV system using such a layout. Machines equipped with IN (OUT) ports only have OUT (IN) ports outside the AGV system boundary. The loads are picked from the OUT ports and delivered to the IN ports. For example, a type D machine is a lift that connects the lower floor to the upper floor of the AGV system, so it has an OUT port only within the AGV system boundary. Each Type A machine has an IN port and an OUT port within the AGV system boundary.

The panels exit machine type A or D to machine type B, and the trays move from machine type C to machine type A. Machine type A receives panels from outside the AGV system and sends them to the AGV system after processing. When a panel exits machine type A, a tray from machine C is required to carry the panel. Machine type D is a lift that transfers panels on trays from the lower floor to the upper floor. In this study, the detailed manufacturing process is not considered; however, the logistics process is simulated by creating delivery requests according to inter-arrival times and their origin and destination locations provided by the company. The AGV characteristics are also provided by the company.

It was assumed that load delivery requests occurred by uniformly distributed inter-arrival times, $U(514, 534)$, at each OUT port. The average number of loads from the OUT ports to the IN ports is presented in Table 2. In the first column, “Panel” represents a panel on a tray. A load delivery request occurred at each loading port, and the unloading port was randomly selected according to the classification in Table 2. The total number of load delivery requests was 3300 in 12 h, that is, a request every 13.1 s for the AGV system on average. The loading capacity of the OUT port was one. Therefore, when a load occurs at an OUT port, if the previous load has not been picked up by an AGV, it is placed immediately after the previous load in the waiting queue of the port and waits until it is picked up.

The AGV system uses unidirectional lines, so AGVs can only move forward. As shown in Figure 2, there are four vertical lines and 13 horizontal lines. The two vertical lines on the left are downward, and the other vertical lines are upward. Six horizontal lines were leftward, and the others were rightward. In the AGV lines, there are IN and OUT port locations of machines and 258 control points (CP), which include intersection points and 1 m interval points on the lines.

The AGV system had 32 AGVs. The moving speed of the AGV was 35 m/min (583.33 mm/s), and its acceleration/deceleration speed was 0.1 m/s^2. Because the capacity of an AGV is one, it can carry one load at a time. The loading (pickup) and unloading (set down) time of an AGV is 15 s. AGVs must stop and turn when they change the moving direction, and the 90 ° turn requires 4.74 s.

In an AGV system, the locations of charging and parking stations are important. However, we did not consider charging operations because they are not directly related to the purpose of this study. We also assumed that idle AGVs circulated along the red AGV lines in Figure 2 instead of parking at specific locations.

Table 2. Average Number of Loads between Machines

Load type	From (# of ports)	To (# of ports)	Count (per 12 hours)
Panel	Machine A (20)	Machine B (24)	1650
Tray	Machine C (12)	Machine A (20)	990
Panel	Machine D (8)	Machine B (24)	660

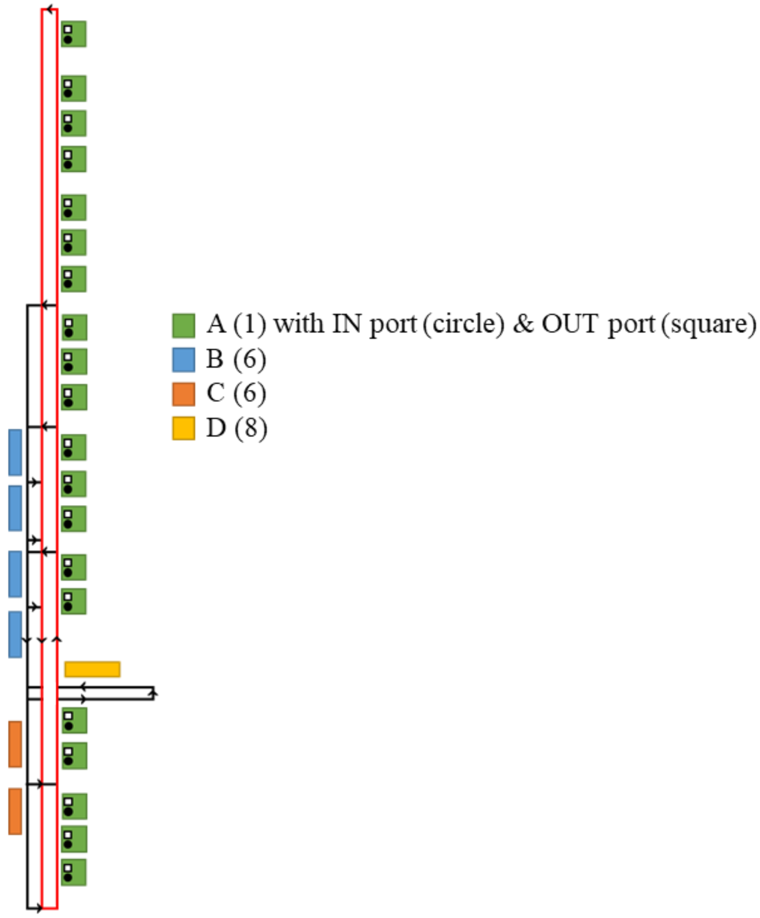

Figure 2. Layout and Location of Machines [Machine Type (Number of Ports per Box)]

4. DISPATCHING AND ROUTING ALGORITHMS

To measure the impact of the dispatching and routing methods, we experimented with three dispatching algorithms and three routing algorithms.

4.1 Dispatching Algorithms

We implemented three dispatching algorithms: the closest vehicle selection rule, RBD, and HABOR. All algorithms were easy to implement, and the computation times were very short. According to the closest vehicle selection rule, when a load delivery request occurs, the idle AGV closest to the loading location of the request is selected if there are idle AGVs. If there is currently no idle AGV, the delivery request remains unassigned and is later assigned to the AGV that completes its current task the fastest. With this rule, once an AGV is assigned to a delivery request, reallocation does not occur.

Using the RBD, the initial allocation of the AGV to a delivery request is performed in the same manner as the closest vehicle selection rule. The difference is that when an AGV becomes idle, and there are no unassigned delivery requests to be assigned to the AGV, the RBD searches for delivery requests that have not been picked up by AGVs and can reduce the waiting time if their currently assigned AGVs are changed to idle AGV. If there are such requests, the delivery request that results in the largest waiting time saving is selected, and the vehicle is reassigned to the idle AGV. In short, an idle AGV can steal a close job if the distance from the already-assigned AGV to a load is longer than that from the idle AGV to that load. For a more detailed description of RBD, refer to Kim *et al*. (2007).

HABOR is an enhanced version of RBD. Whereas an idle AGV takes a near delivery request, and there is at most one reassignment at a time by the RBD, multiple reassignments are possible by HABOR. When an AGV becomes idle, HABOR is performed when a certain period of time has elapsed from the previous HABOR. HABOR searches all the AGVs that do not carry loads, including the AGVs moving to retrieve loads and all the delivery requests whose loads have not been picked up. The distances between AGVs and loads are calculated as an assignment cost, and an assignment problem is formulated and solved using the Hungarian algorithm. Therefore, multiple reassignments occurred. HABOR reduces the total lead time by reallocating multiple vehicles and loads concurrently. Although HABOR has been proposed for OHT systems, it can be applied to AGV systems without modification. Detailed descriptions of the HABOR are provided by Kim *et al*. (2009).

4.2. Routing Algorithms

We implemented three routing algorithms: the shortest route selection rule, the dynamic A* algorithm, and the advanced dynamic A* algorithm. The shortest route selection rule was the default selection method applied by AutoMod. Based on this rule, the shortest route from one location to another is selected, and an AGV follows this route. The A* algorithm is a well-known shortest-path search algorithm for one-to-one paths. The other two algorithms are described below.

4.2.1 Dynamic A* Algorithm

We used the dynamic A* algorithm to determine the fastest route from the current position to the destination. This routing algorithm can navigate and change the route while an AGV is moving instead of maintaining to the initial route path calculated when starting the movement in the traditional A* algorithm method. In addition, our dynamic A* algorithm takes into consideration the turn penalty owing to the grid layout.

To calculate the travel time of a route, we used W value as the estimated travel time for each link between any two adjacent CPs. At first $W_{init}(i,j)$ is set to the value of distance between CPs i and j divided by the AGV speed, as shown in Equation (3). Whenever an AGV passes the link between CPs i and j, $W(i,j)$ is updated by exponential smoothing with parameter α, as shown in Equation (4). Using this procedure, $W(i,j)$ stores experienced information. On the contrary, the shortest route selection rule uses $W_{init}(i,j)$ only. In the experiments, we set the value of α to 0.5 after preliminary experiments.

$$d(i,j) = \text{ distance from CP } i \text{ to CP } j \tag{1}$$
$$t(i,j) = \text{ AGV travel time from CP } i \text{ to CP } j \tag{2}$$
$$W_{init}(i,j) = \frac{d(i,j)}{speed\ of\ AGV} \tag{3}$$
$$W_{new}(i,j) = W(i,j) + \alpha * \left(t(i,j) - \mathrm{W}(i,j)\right) \tag{4}$$

As the AGV system in this study requires right-angle turns, the turn times of a route must be considered if the route contains turns. Unlike an AGV system with a curved-line structure, an AGV system with a grid-line structure has only straight lines. Therefore, when an AGV moves from a horizontal line to a vertical line, it is necessary to stop, rotate, and start to move again. These activities require additional time and must be considered when the travel time from a location to a location is calculated by the A* algorithm. Accordingly, we introduce a penalty for each turn.

In the A* algorithm, the travel time from the start location to the destination location through CP i, $f(i)$, is estimated using $g(i)$ plus $h(i)$. $g(i)$ is the travel time from the start location to CP i, and is calculated using the estimated link travel times W_{new}. When $g(i)$ is calculated, the number of turns of the route is counted, and the corresponding turn penalty is added. $h(i)$ is the lower bound of travel time from CP i to the destination location. Assuming that there is no AGV between CP i and the destination location, we used the initial link travel times W_{init} considering the turn times to calculate $h(i)$. Thus, $h(i)$ value is static, and the lower bounds of the travel times from all CPs to all CPs can be precalculated at the beginning.

Figure 3 shows the two routes between the two locations. If the turn penalty is not counted, the travel times of the routes will be the same if the link travel times are proportional to their lengths. However, with the turn penalty, Route 1 is faster than Route 2 because the former has one turn, whereas the latter has two. Because an AGV must stop for turning, its actual travel time is increased due to its deceleration and acceleration at turning. Thus, the turn penalty is set to a larger value than the actual turn time. In our experiments, we set *Turn Penalty* to 10 s, whereas it takes 4.74 s for an AGV to turn 90 °, as described in Section 3.

Using the dynamic A* algorithm, each AGV selects the shortest route to its destination at each CP and dynamically changes its path while moving. When an AGV searches for the shortest-time route using the A* algorithm at each CP, the estimated travel times between CPs are used, and they are continuously updated by the movements of the AGVs and exponential smoothing, as shown in Equation (4).

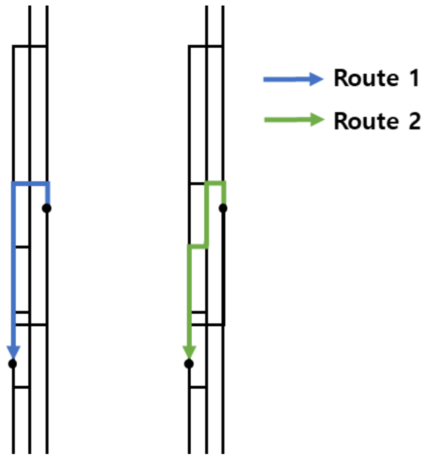

Figure 3. Examples of a Route with Turn Penalty and a Route without Turn Penalty

4.2.2 Advanced Dynamic A* Algorithm

Although A* algorithms have been widely used, as summarized in Section 2, we have found some weaknesses in the algorithm applied to our AGV system. The dynamic A* algorithm uses $W(i,j)$ to determine the fastest route to the target destination of an AGV. However, updating $W(i,j)$ with only the exponential smoothing method resulted in the problems described below. To solve this problem, we proposed two methods of refreshing and restricting the upper limit.

The major problem with the algorithm is that $W(i,j)$ is updated only when an AGV passes the link (CP *i* to CP *j*). Suppose that some AGVs have passed (i,j) and experienced heavy traffic jams. Then, $W(i,j)$ was set to a large value using Equation (4). This large value prevents the selection of the link unless other links from CP *i* have similarly large values. Then, the link is no longer selected as part of the route. However, in many cases, the link's traffic jam is soon resolved, and there is no AGV on the link. To address this updating problem, we propose refreshing $W_{new}(i,j)$ using exponential smoothing with parameter α and the initial value $W_{init}(i,j)$, as expressed in Equation (5). When the value of a link is not updated for a certain period of time, the value is refreshed by the Equation. The threshold period was set to 30 s.

$$W_{new}(i,j) = W(i,j) + \alpha * \left(W_{init}(i,j) - \mathrm{W}(i,j)\right). \tag{5}$$

The second problem of the dynamic A* algorithm is similar to the first. When some AGVs experience heavy congestion on (i,j) link, $W_{new}(i,j)$ can be set to a very large value, and the link is not selected as a route link. In some cases, this causes an inefficient route selection. Figure 4 shows an example of this phenomenon. When the middle short link has a very large value of $W(i,j)$, the surrounding path can be selected as shown in the left figure. However, the path around cannot be more efficient than the short link shown in the right figure because of the line network structure. To prevent AGVs from moving far away, we introduced an upper limit of the estimated time on each link, as shown in Equation (6). Now, the travel time is updated using Equation (7). The value of β was set to 15.

$$W_{max}(i,j) = W_{init}(i,j) * \beta \tag{6}$$
$$W_{new}(i,j) = min\,(W(i,j) + \alpha * \left(t(i,j) - \mathrm{W}(i,j)\right), W_{max}(i,j)). \tag{7}$$

We call this modified version, which has a refresh mechanism with Equation (5) and uses Equation (7) for updating, *advanced dynamic A* algorithm*.

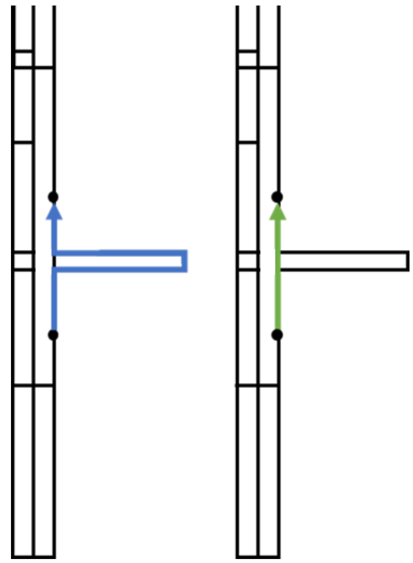

Figure 4. Examples of a Route without Bounded Maximum and a Route with Bounded Maximum.

5. SIMULATION RESULTS

All simulation experiments were conducted using AutoMod (package version 12.5.2). This software has been used in many AMHS-related studies, including Kim *et al*. (2007), Kim *et al*. (2009), and Hwang and Jang (2020). AutoMod provides two moving vehicle systems for modelling AGV/OHT: the path mover system and the power and free system. Although both systems can be used in our experiment, the path mover system was selected in this study.

We used 2 hours warm-up, 12 hours W value initial update, and 12 hours result collection. Thus, each simulation run lasted 26 h. Twenty replications were performed for each simulation setting.

5.1 Results of Dispatching and Routing Algorithms with 32 AGVs

The experiment was conducted using 32 AGVs. We performed nine combinatorial experiments using three dispatching and three routing algorithms. The parameters used in the experiment are shown in Table 3.

Table 4 lists our simulation results. The columns labeled 'Retrieve' show the retrieval time, which is the duration between the load delivery request occurrence and the pick-up time of the load. The columns labeled 'Delivery' show the delivery time, which is the duration from a load pick-up time to its unloading time at its destination. The columns labeled 'Total' indicate the total lead-time from the load delivery request occurrence to the unloading time. All times are in seconds. The rows labeled 'Avg.' and 'Std' shows the average and standard deviation of times, respectively. The simulated dispatching algorithms are closest vehicle selection rule (labeled 'Closest'), RBD, and HABOR, and the routing algorithms are the shortest route selection rule (labeled 'Shortest'), a dynamic A* algorithm (labeled 'Dynamic A*'), and an advanced dynamic A* algorithm (labeled 'Advanced Dynamic A*'). The shortest routing is static, and the route from a location to a location is not changed, whereas the other two routings are dynamic, and the route can be changed. The number of completed delivery requests was not explicitly reported, as all delivery requests were completed in all the combinatorial experiments.

Table 3. Simulation Parameters

Parameter	Value
α	0.5
β	15
Turn penalty (s)	10
Refresh threshold (s)	30

Table 4. Simulation Results

Dispatching \ Routing		Shortest Retrieve (s)	Shortest Delivery (s)	Shortest Total (s)	Shortest AGV Util.(%)	Dynamic A* Retrieve (s)	Dynamic A* Delivery (s)	Dynamic A* Total (s)	Dynamic A* AGV Util.(%)	Advanced Dynamic A* Retrieve (s)	Advanced Dynamic A* Delivery (s)	Advanced Dynamic A* Total (s)	Advanced Dynamic A* AGV Util.(%)
Closest	Avg.	292.7	182.7	475.4	92.5	286.3	179.3	465.6	91.1	287.7	177.3	464.9	90.0
	Std.	5.9	2.1	5.9	0.9	6.9	2.0	6.7	1.0	6.4	1.7	6.2	0.9
RBD	Avg.	188.5	178.9	367.3	79.2	180.2	174.4	354.6	77.5	177.8	173.0	350.8	76.7
	Std.	9.3	2.2	10.3	1.8	6.9	2.2	7.6	1.3	7.6	2.6	8.0	1.4
HABOR	Avg.	160.6	180.5	341.1	79.0	161.1	175.3	336.3	78.3	157.4	173.6	331.1	77.3
	Std.	10.0	2.3	11.1	1.8	9.5	2.6	10.3	1.7	9.0	1.8	9.5	1.7

The impact of the dispatching algorithms is significant. With the basic routing algorithm (Shortest), RBD and HABOR reduced the total lead time by 22.7 % (from 475.4 s to 367.3 s) and 28.2 % (from 475.4 s to 341.1 s), respectively, compared to the ‘Closest’ algorithm. With the dynamic A* routing algorithm, RBD and HABOR could also reduce the total lead time by 23.8 % and 27.8 %, respectively. With the advanced dynamic A* routing algorithm, RBD and HABOR could also reduce the total lead time by 24.6 % and 28.8 %, respectively.

The dispatching algorithm effectively reduced the retrieval time, which is the time taken to reach the pickup location of the delivery request, as the assigned load vehicle changes to a nearby vehicle. The retrieval time depends on which vehicle is assigned, hence dispatching methods affect the change in retrieval time. With the shortest routing algorithm, RBD and HABOR reduced the retrieval time by 35.6 % (from 292.7 s to 188.5 s) and 45.1 % (from 292.7 s to 160.6 s), respectively, compared to the ‘Closest’ algorithm. With the dynamic A* routing algorithm, RBD and HABOR can also reduce the retrieval time by 37.1 % and 10.6 %, respectively. With the advanced dynamic A* routing algorithm, RBD and HABOR can also reduce the retrieval time by 38.2 % and 11.5 %, respectively. As a result, the dispatch algorithms effectively reduce the retrieval time through effective vehicle allocation.

The impact of routing algorithms is meaningful but not as significant as that of dispatching algorithms. With the basic dispatching algorithm (Closest), dynamic A* and advanced dynamic A* could reduce the total lead time by 2.1 % (from 475.4 s to 465.6 s) and 2.2 % (from 475.4 s to 464.9 s), respectively, compared to the ‘Shortest’ algorithm. With the RBD dispatching algorithm, dynamic A* and advanced dynamic A* could also reduce the total lead-time by 3.5 % and 4.5 %, respectively. With the HABOR dispatching algorithm, dynamic A* and advanced dynamic A* could also reduce the total lead-time by 1.4 % and 3.0 %, respectively. The advanced dynamic A* algorithm clearly decreases the lead time and AGV utilization over static shortest routing. It also shows improvement effects over dynamic A* algorithm.

Since the routing algorithms change the travel path of AGVs, it directly affects the delivery time, although the start and end positions are the same. With the closest dispatching algorithm, dynamic A* and advanced dynamic A* can reduce the delivery time by 1.9 % (from 182.7 s to 179.3 s) and 3.0 % (from 182.7 s to 177.3 s), respectively, compared to the ‘Shortest’ algorithm. With the RBD dispatching algorithm, dynamic A* and advanced dynamic A* can also reduce the delivery time by 2.5 % and 3.3 %, respectively. With the HABOR dispatching algorithm, dynamic A* and advanced dynamic A* can also reduce the delivery time by 2.9 % and 3.8 %, respectively. As a result, routing algorithms reduced the total lead time by reducing the delivery time.

The impact of the simultaneous application of dispatching and routing algorithms was significant. The application of HABOR and advanced dynamic A* resulted in a 30.4 % reduction in the average total lead time (from 475.4 s to 331.1 s).

We further investigated the impact of the integration of dispatching and routing algorithms. Table 5 shows the t-values of the paired t-test for the average total lead-time. For the paired t-test, we assumed the following hypotheses:

- H_0: The average lead time of the row algorithm is not different from that of the column algorithm.
- H_A: The average lead time of the column algorithm is lower than that of the row algorithm.

Therefore, the large value in Table 5 indicates an improvement in the column algorithm over the row algorithm. If the improvement is statistically significant with a p-value of less than 0.05, the values are marked in bold. All comparisons were considered statistically significant.

The simulation results showed that Advanced Dynamic A* can reduce the lead time, but the impact of dispatching algorithms, such as RBD and HABOR, is more significant. When a load delivery request occurs, the retrieval distance between the load and AGV varies significantly depending on which AGV is selected. Thus, the impact of dispatching decisions is significant. On the other hand, although the delivery time depends on which route is selected, there is a limit to reducing the routing time because the traveling distances among the possible routes are similar. In spite of the fact, a smart routing decision can still reduce the lead time. In summary, effective dispatching and routing algorithms must be used together to realize a smart AHMS.

Table 5. Paired t-test Statistics (t-value) of Average Total Lead Time between Integrated Algorithms

Integrated Algorithms (Dispatching + Routing)	1	2	3	4	5	6
1. Closest + Shortest		**43.58**	**56.70**	**7.83**	**61.11**	**69.51**
2. RBD + Shortest	-43.58		**8.43**	-39.18	**8.28**	**12.80**
3. HABOR + Shortest	-56.70	-8.43		-59.69	-3.35	**8.13**
4. Closest + Advanced Dynamic A*	-7.83	**39.18**	**59.69**		**56.58**	**76.90**
5. RBD + Advanced Dynamic A*	-61.11	-8.28	**3.35**	-56.58		**7.70**
6. HABOR + Advanced Dynamic A*	-69.51	-12.80	-8.13	-76.90	-7.70	

5.2 Simulation Results with Different Numbers of AGVs

In this section, the effects of the algorithms according to the number of AGVs are analyzed. Simulations were conducted by adjusting the number of AGVs from 29 to 35.

Table 6. Simulation Results According to the Number of AGVs

Number of AGVs	Total lead time (s)						AGV Utilization (%)					
	1	2	3	4	5	6	1	2	3	4	5	6
29	457.5	376.7	358.2	447.4	358.2	349.2	85.93	78.15	77.85	84.22	75.79	76.60
30	463.2	371.5	349.7	453.7	354.5	340.3	88.17	78.41	78.13	86.25	76.16	76.66
31	469.3	369.6	345.0	458.6	353.3	333.2	90.34	79.04	78.60	88.20	76.71	76.69
32	475.4	367.3	341.1	464.9	350.8	331.1	92.46	79.24	79.04	90.00	76.66	77.33
33	481.8	363.4	339.2	472.1	350.8	328.6	94.40	79.01	79.64	92.01	77.09	77.72
34	489.8	364.8	339.8	478.6	351.5	328.1	96.55	79.36	80.49	93.80	77.34	78.21
35	499.2	365.2	336.8	488.1	355.1	326.1	98.83	79.51	80.43	96.24	78.03	78.26

* The algorithm indices 1, 2, 3, 4, 5, and 6 are listed in Table 5.

Table 6 shows the simulation results, and Figures 5 and 6 show the graphs of the total lead time and AGV utilization according to the number of AGVs, respectively. The 'Total lead time' column represents the average value of the total lead time in seconds, from load delivery request occurrence to the unloading time, and ‘AGV Utilization’ represents the average value of AGV utilization for each algorithm. The index of the algorithm is listed in Table 5.

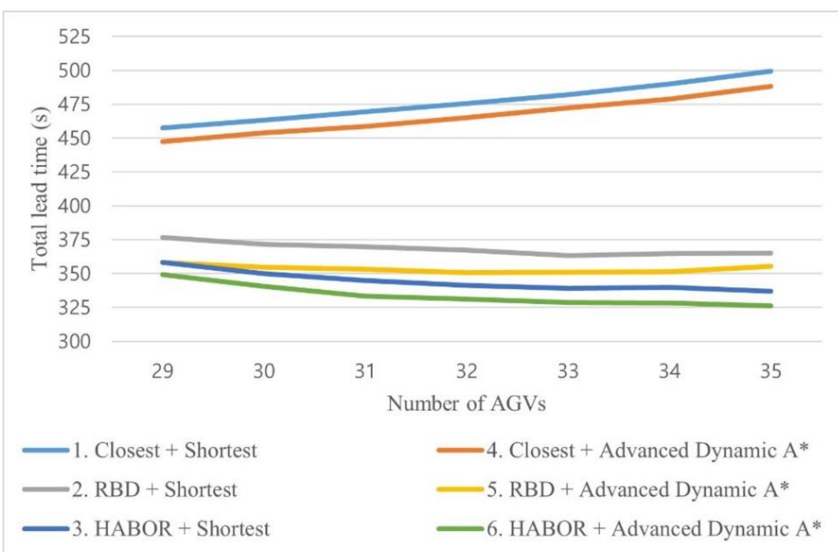

Figure 5. Total Lead Time According to the Number of AGVs

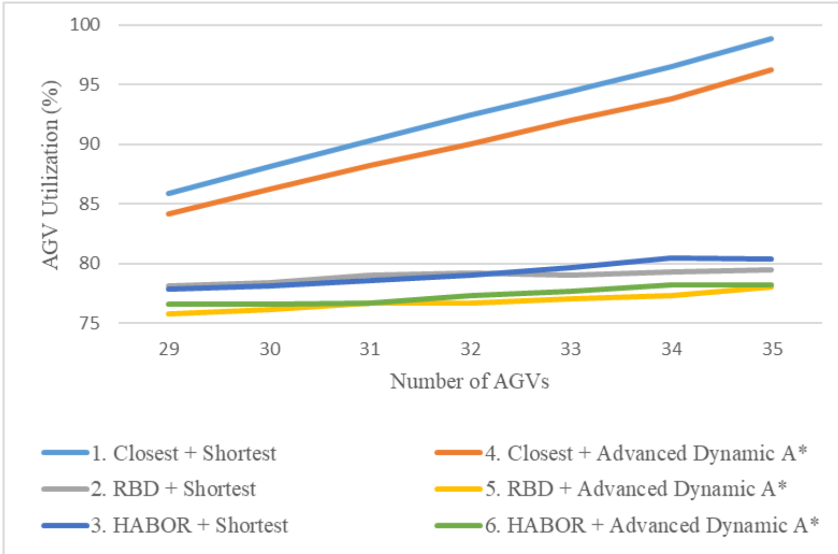

Figure 6. Utilization According to the Number of AGVs

Based on the simulation results, we can conclude that the impact of dispatch rules and routing algorithms is consistently significant, regardless of the number of AGVs. As the number of AGVs increases, the total lead times and utilization of the Closest dispatching rule increase. This implies that the more AGVs there are, the more complex the traffic flow. However, with RBD and HABOR, as the number of AGVs increases, the total lead time decreases. The two dispatching algorithms allow the AGV system to accommodate a larger number of AGVs without heavy traffic congestion by efficiently assigning loads to AGVs. Additionally, the advanced dynamic A* routing algorithm can reduce the total lead times, regardless of the number of AGVs.

5.3 Simulation Results with Different Load Occurrences

In this section, the effects of the algorithms according to the number of loads (delivery requests) are analyzed. Simulations were conducted by adjusting the number of loads with 32 AGVs. Table 7 shows the simulation results; the retrieval, delivery, total lead times, and AGV utilization according to the number of loads.

Table 7. Simulation Results with Different Number of Loads

Number of Loads	Routing / Dispatching		Shortest				Advanced Dynamic A*			
			Retrieve (s)	Delivery (s)	Total (s)	AGV Util.(%)	Retrieve (s)	Delivery (s)	Total (s)	AGV Util.(%)
2100	Closest	Avg.	309.1	170.8	479.9	58.8	295.7	165.5	461.2	57.3
		Std.	8.8	5.9	14.7	1.0	8.9	3.9	12.8	0.6
	RBD	Avg.	245.1	169.2	414.3	53.0	243.3	166.0	409.3	53.1
		Std.	0.1	3.5	3.6	0.4	7.6	1.4	6.2	2.0
	HABOR	Avg.	197.0	170.7	367.7	53.3	192.6	169.0	361.6	52.7
		Std.	0.3	6.4	6.1	0.3	2.9	3.8	6.8	0.2
3600	Closest	Avg.	282.3	190.5	472.8	98.4	268.7	180.8	449.5	96.4
		Std.	8.7	1.8	8.6	0.5	7.9	2.7	7.4	0.5
	RBD	Avg.	251.4	186.8	438.1	95.7	228.6	177.6	406.2	90.9
		Std.	10.2	2.0	8.5	0.5	9.4	1.7	9.3	1.2
	HABOR	Avg.	211.4	188.3	399.7	92.5	188.1	180.3	368.4	88.0
		Std.	13.0	1.0	12.8	1.4	15.3	0.9	15.8	1.7

The impact of the dispatching algorithms is significant in both number of loads cases. With 2100 loads per 12 h using the shortest routing algorithm, RBD and HABOR reduced the total lead time by 13.7 % (from 479.9 s to 414.3 s) and 23.4 %

(from 479.9 s to 367.7 s), respectively, compared to the 'Closest' algorithm. Using the dynamic A* routing algorithm, RBD and HABOR reduced the total lead time by 11.3 % and 21.6 %, respectively. With 3600 loads, RBD and HABOR reduced the total lead time by 7.3% and 15.5% with shortest routing, respectively, and 9.6% and 18.0% with the dynamic A* routing algorithm, respectively. As discussed in Section 5.1, the retrieval time decreases significantly according to the dispatching algorithms. When the vehicle utilization is lower, the impact of the dispatching algorithms is bigger.

The impact of the routing algorithms is also noticeable in both cases. With 2100 loads per 12 h and three dispatching algorithms, the advanced dynamic A* can reduce the total lead time by 3.9 % (from 479.9 s to 461.2 s), 1.2 % (from 414.3 s to 409.3 s), and 1.7 % (from 367.7 s to 361.6 s), respectively, compared to the shortest algorithm. With 3600 loads, it reduced by 4.9 %, 7.3 %, and 7.8 %, respectively. As discussed in Section 5.1, the routing algorithms reduce the delivery time slightly. When the vehicle utilization is higher, the impact of the routing algorithms is bigger.

Based on the simulation results, we can conclude that the impact of the dispatching and routing algorithms is consistently large regardless of the number of loads, although dispatching methods show a more significant impact.

6. CONCLUSIONS

In this study, we experimented with three dispatching and three routing algorithms and measured their impact on the total lead time using a simulation model for an AGV system. The dispatching algorithms are the closest vehicle selection rule, RBD, and HABOR, and the routing algorithms are the shortest route selection rule, dynamic A^* algorithm, and advanced dynamic A^* algorithm, respectively. The proposed dynamic A* algorithms determine the fastest routes with travel time estimation. For the routing algorithms, we considered the AGV turn time owing to the grid-based layout structure. We also found some weaknesses of the A* algorithm applied to our AGV system and proposed an advanced dynamic A* algorithm by introducing an upper bound limit on link estimated time and a refresh method. The productivity improvements (decrease in the average lead time) of the advanced dynamic A* algorithm were verified through simulation experiments.

The impact of a good dispatching algorithm is significant (ranging from 27.8 % to 28.8 % of lead time reduction). The impact of routing algorithms is also meaningful, but not as significant as that of dispatching algorithms. The simultaneous integration of the dispatching rule and routing algorithm showed the most effective reduction in total lead time. The application of HABOR and Advanced Dynamic A* resulted in a 30.4 % reduction in the average total lead-time. We conclude that effective dispatching and routing algorithms must be used together to realize a smart AHMS. This study can be considered as a case study; however, the findings of this study are general and can be applied to other environments.

There are opportunities to improve or extend this study. First, our experiments were limited to an AGV system. Similar experiments can be performed for other AMHS with different layout structures and production environments. Second, more sophisticated routing algorithms can be developed. The proposed dynamic A* algorithms uses only historical data to estimate the link travel times. However, additional information, such as the current positions of the AGVs and their planned movement schedule, can be used to estimate future travel times more accurately. Third, research on dispatching and routing algorithms in volatile environments can be conducted. For example, load delivery request occurrence patterns, consisting of locations and quantities, may change dynamically. Fourth, this study considered unit-load AGVs. This research can be extended to AGV systems with multi-load AGVs.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2C2006665 and No. 2022R1A4A1019071).

REFERENCES

Bae, J., and Chung, W. (2019). Efficient Path Planning for Multiple Transportation Robots under Various Loading Conditions. *International Journal of Advanced Robotic Systems*, 16 (2): 1–9.

Bartlett, K., Lee, J., Ahmed, S., Nemhauser, G., Sokol, J., and Na, B. (2014). Congestion-Aware Dynamic Routing in Automated Material Handling Systems. *Computers and Industrial Engineering*, 70 (1): 176–182.

Chawla, V.K., Chanda, A.K., and Angra, S. (2019). Automatic Guided Vehicle Systems in the Flexible Manufacturing System – a Review. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 26(5): 737-765.

Corréa, A. I., Langevin, A., and Rousseau, L. (2007). Scheduling and Routing of Automated Guided Vehicles: A Hybrid Approach. *Computers and Operations Research*, 34(6): 1688-1707.

Dai, X., Long, S., Zhang, Z., and Gong, D. (2019). Mobile Robot Path Planning Based on Ant Colony Algorithm with A* Heuristic Method. *Frontiers in Neurorobotics*, 13. https://doi.org/10.3389/fnbot.2019.00015.

Desaulniers, G., Langevin, A., Riopel, D., and Villeneuve, R. (2003). Dispatching and Conflict-Free Routing of Automated Guided Vehicles: An Exact Approach. *International Journal of Flexible Manufacturing Systems*, 15(4): 309-331.

Fransen, K., and Van Eekelen, J. (2021). Efficient Path Planning for Automated Guided Vehicles Using A* (Astar) Algorithm Incorporating Turning Costs in Search Heuristic. *International Journal of Production Research*, (In press). https://doi.org/10.1080/00207543.2021.2015806.

Fransen, K.J.C., Van Eekelen, J.A.W.M., Pogromsky, A., Boon, M.A.A., and Adan, I.J.B.F. (2020). A Dynamic Path Planning Approach for Dense, Large, Grid-Based Automated Guided Vehicle Systems. *Computers and Operations Research*, 123: 1–10.

Ganesharajah, T., Hall, N.G., and Sriskandarajah, C. (1998). Design and Operational Issues in AGV-Served Manufacturing Systems. *Annals of Operations Research*, 76: 109-154.

Hart, P.E., Nilsson, N.J., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4 (2): 100–107.

Hwang, I., and Jang, Y. (2020). Q(λ) Learning-based Dynamic Route Guidance Algorithm for Overhead Hoist Transport Systems in Semiconductor Fabs. *International Journal of Production Research*, 58(4): 1199-1221.

Ki, Y., Na, B., and Kim, B. (2019). Travel Time Prediction-Based Routing Algorithms for Automated Highway Systems. *IEEE Access*, 7: 121709-121718.

Kim, B., Shin, J., Jeong, S., and Koo, J. (2009). Effective Overhead Hoist Transport Dispatching Based on the Hungarian Algorithm for a Large Semiconductor FAB. *International Journal of Production Research*, 47 (10): 2823-2834.

Kim, B., Oh, S., Shin, J., Jung, M., Chae, J., and Lee, S. (2007). Effectiveness of Vehicle Reassignment in a Large-Scale Overhead Hoist Transport System. *International Journal of Production Research*, 45: 789-802.

Lee, S., Lee, J., and Na, B. (2018). Practical Routing Algorithm Using a Congestion Monitoring System in Semiconductor Manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 31(4): 475-485.

Martins, O.O., Adekunle, A.A., Olaniyan, O.M., and Bolaji, B.O. (2022). An Improved Multi-Objective A-Star Algorithm for Path Planning in a Large Workspace: Design, Implementation, and Evaluation. *Scientific African*, 15: e01068.

Mu, T., Zhu, J., Li, X., and Li, J. (2020). Research on Two-Stage Path Planning Algorithms for Storage Multi-AGV. *Communications in Computer and Information Science*, 1160: 418–430.

Nazzal, D., and Bodner, D.A. (2003). A Simulation Based Design Framework for Automated Material Handling Systems in 300mm Fabrication Facilities. *Proceedings of the 2003 Winter Simulation Conference*, New Orleans, LA, USA, December 7-10, 1351-1359.

Nishi, T., Hiranaka, Y., and Grossmann, I.E. (2011). A Bilevel Decomposition Algorithm for Simultaneous Production Scheduling and Conflict-Free Routing for Automated Guided Vehicles. *Computers and Operations Research*, 38(5): 876-888.

Raajan, J., Srihari, P.V., Satya, J.P., Bhikkaji, B., and Pasumarthy, R. (2020). Real Time Path Planning of Robot using Deep Reinforcement Learning. *IFAC-PapersOnLine*, 53(2): 15602-15607.

Tang, B., Zhu, Z., and Luo, J. (2016). Hybridizing Particle Swarm Optimization and Differential Evolution for the Mobile Robot Global Path Planning. *International Journal of Advanced Robotic Systems*, 13 (3): 86.

Tuncer, A., and Yildirim, M. (2012). Dynamic Path Planning of Mobile Robots with Improved Genetic Algorithm. *Computers and Electrical Engineering*, 38 (6): 1564-1572.

Yuan, Z., Yang, Z., Lv, L., and Shi, Y. (2020). A Bi-Level Path Planning Algorithm for Multi-AGV Routing Problem. *Electronics*, 9(9): 1351.

Zhao, J., and Fu, X. (2012). Improved Ant Colony Optimization Algorithm and Its Application on Path Planning of Mobile Robot. *Journal of Computers*, 7(8): 2055-2062.