ROUTING AGVS IN CONTAINER TERMINALS BY USING Q-LEARNING

Su Min Jeon*, Kap Hwan Kim* and Herbert Kopfer **

[*] *Department of Industrial Engineering, Pusan National University, Jangjeon-dong, Kumjeong-ku, Busan 609-735, Korea.*

e-mail: {1006sumin, kapkim}@pusan.ac.kr

[**] *Lehrstuhl für Logistik, FB 7, University of Bremen, D-28334 Bremen, Germany.*

e-mail: kopfer@uni-bremen.de

ABSTRACT

This paper suggests a routing method for automated guided vehicles (AGVs) in port terminals by using a Q-learning technique. One of the important issues for the efficient operation of automated guided vehicle system (AGVS) is to find the shortest time route instead of the shortest distance route which is usually being used in practice. For the estimation of the travel time, the waiting time must be estimated accurately. This study proposes a method for estimating the waiting time of vehicles resulting from the interference among vehicles during the travel by using the Q-learning technique. An experiment was performed to evaluate the performance of the learning algorithm by a simulation technique. The performance of the learning-based routes was compared to that of the shortest distance routes by the simulation study.

Key Words: AGV, reinforcement learning, routing, container terminal

1. INTRODUCTION

There are four types of operations (unloading, loading, receiving, and delivery) performed by handling equipment in port container terminals. We assume that three types of equipment are used for ship operations such as QC (Quay Crane), AGV (Automated guided vehicle), and AYC (Automated yard crane). For example, the unloading operation can be decomposed into 3 steps. The first step is performed by a QC transferring a container from a ship and putting down the container on an AGV. The second step is performed by an AGV delivering the container from the QC to a block in the storage yard. The last step is performed by an AYC picking up the container and stacking it into a position in a yard block.

For the operation of AGVs, the control system must have such functions as dispatching, routing or scheduling, and traffic control. The routing function of vehicles is one of the most important components of operational control systems to achieve a high productivity in automated container terminals. The routing function selects a specific path for a vehicle to follow to reach its destination from the present position.

Usually, a vehicle is given a predetermined route from its starting position to its destination. Considering the efficiency of the terminal operation, the shortest distance routes are usually provided to vehicles. This paper attempts to find the shortest time route instead of the shortest distance route by considering the congestion at intersections and bidirectional path segments. To find the shortest travel time route, the expected delay time of vehicles at each intersection is estimated by utilizing travel experiences of vehicles collected during a simulation process.

The AGV routing problem has been addressed by several researchers. The conceptual foundations of the AGV routing problem were first laid by Broadbent et al (1985). They proposed an AGV scheduler that uses Dijkstra's shortest path algorithm and generates a timetable containing the node occupation times for each vehicle. A study by Gaskins and Tanchoco (1987) suggested an integer programming model to determine the directions of path segments on guide path in a way of minimizing the total travel distance of vehicles. Kim and Tanchoco (1991) used the concept of time window graph, which is a directed graph of the free time windows, for finding the shortest time route on bi-directional guide path networks. Rajotia *et al*. (1998) proposed a semi-dynamic time window routing strategy, the principle of which is quite similar to the path planning method of Kim and Tanchoco (1991). Time windows modeling the traffic flow direction are placed on bi directional arcs, which can only be crossed according to one direction at a time. Based on these time windows, the Dijkstra algorithm was applied to find the least congested and fastest routes for vehicles. Oboth *et al*. (1999) addressed operational control problems such as the demand assignment and the route planning. They proposed a route generation procedure called the sequential path generation (SPG) heuristic. Lim *et al*. (2000) applied a Q-learning method (Mahadevan 1996, Mitchell 1997) to estimate the expected travel time of vehicle on path segments for designing guide paths in AGVSs.

This study also applies the Q-learning method for a route planning for AGVs in automated container terminals. For finding the shortest time route for a vehicle from its starting location to a final destination, the schedule has to ensure a conflict and deadlock free travel during the entire travel. Thus, this study addressed not only the route construction method but also a traffic control issue for resolving the deadlocks during the travel of the vehicle.

The rest of this paper is structured as follows. In section 2, we introduce a guide path network for AGVs assumed in this study and the traffic control problem. Section 3 describes how to apply the Q-leaning method to make the shortest time routes. Section 4 compares the system performance between routes by the learning method and the shortest distance route by a simulation study. Finally, the conclusion and summary are provided in section 5.

2. AGV PATH NETWORK AND THE ROUTING PROBLEM

Figure1 illustrates a typical logistics process for unloading operation in automated container terminals. In automated container terminals (ACT), free-ranging AGVs are usually used, which do not have physically permanent guide paths constructed by using electric wires or magnetic tapes and thus can travel on a guide path network temporarily specified in the memory of the supervisory control computer. Two types of the guide path network used in practice are of closed-loop type and of cross-lane type. The guide path network of the closed-loop type has several large circular guide paths for vehicles follow for the travel. Thus, the guide path network of the closed-loop type allows a simplified control of vehicles but requires long travel distances for vehicles.

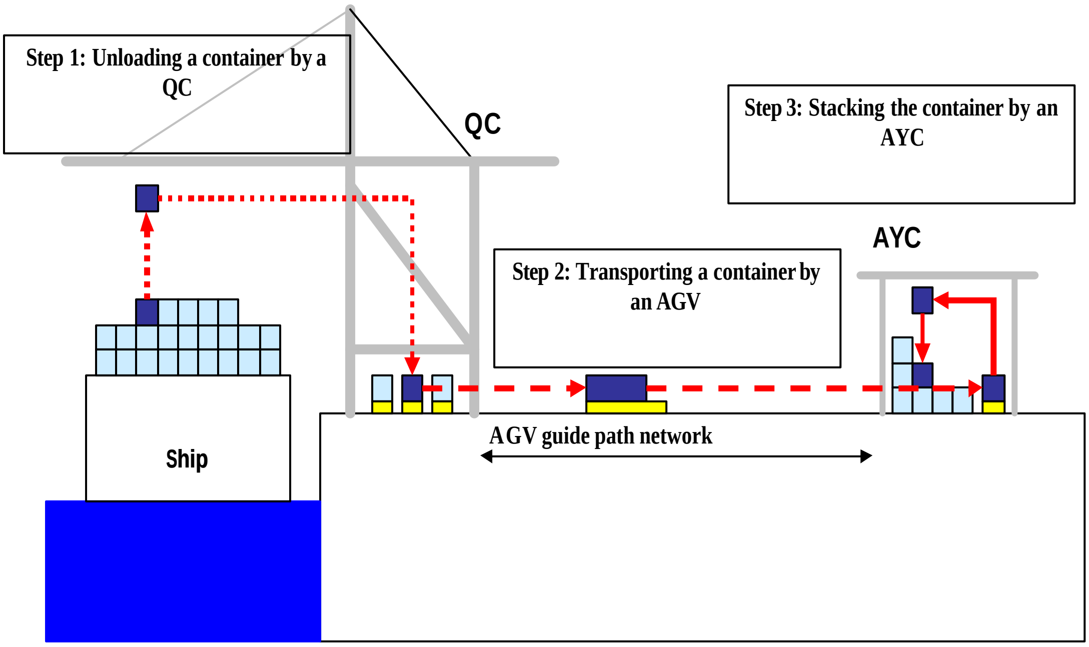
Step 1: Unloading a container by a QC
QC
Step 3: Stacking the container by an AYC
AYC
Step 2: Transporting a container by an AGV
Ship
AGV guide path network

Fig1. An illustration of unloading operation

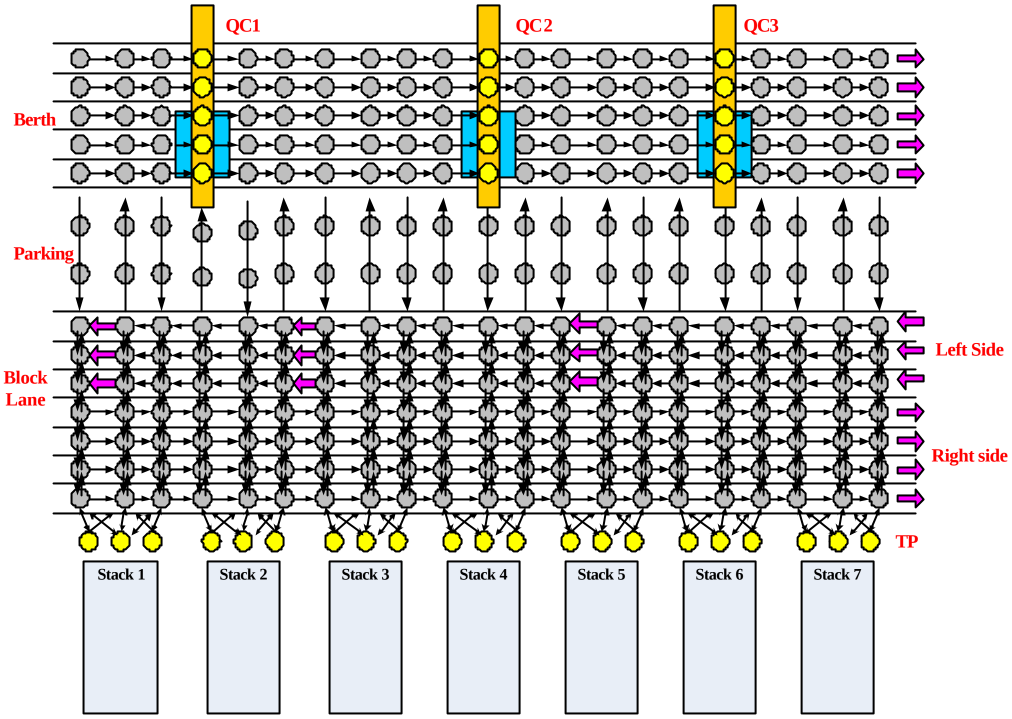

Fig 2. An illustration of a guide path network

To speed up the deliveries of containers by AGVs, more complicated guide path networks have been applied in ACTs. On the cross-lane guide path networks, which are being used in practice, a vehicle moves on shortcuts to travel on the shortest possible route from its origin to its destination. Although cross-lane guide path networks can reduce the travel distances of AGVs significantly, the traffic control of vehicles becomes much more complicated.

This study assumes the guide path network of the crossed lane type as illustrated in Fig2. Fig2 shows a layout of the guide path network with yard blocks laid out in the perpendicular to the berth. In the network, a node corresponds to an intersection point of path segments or pickup and delivery (P/D) points for QCs or blocks. We use the zone control policy for preventing the collision among AGVs. An arc in the network represents the traveling direction allowed for AGVs. Defining the travel direction on each arc is an important decision-making for the design of the guide path network. We assume that there are 5 traveling lanes under QCs including 3 lanes allocated for transferring containers with QCs. The other two lanes are used only for the running of vehicles. All the lanes under QCs are directed toward the same direction. Some lanes in the parking area are directed toward the berth, while some lanes are directed toward the yard. There are 7 lanes for the running of vehicles in front of blocks and each block has 3 transfer points (TPs).

3. APPLICATION OF THE Q-LEARNING TECHNIQUE FOR AGV ROUTING

This section introduces how to plan the routes for vehicles by Q-learning technique. The objective is to get routes for minimizing the travel time for a given starting location and the destination.

Reinforcement learning is a process of learning how to match situations with actions in order to maximize a numerical reward signal. The learners are not told what actions to take, as in the case of most of machine learning algorithms. Instead, learners must discover by trial and error what actions yield the most reward. The four sub-elements of reinforcement learning are policy, reward function, value function, and model of environment. The following describes how the Q-learning technique can be applied to the routing of vehicles (Lim et. al, 2002).

In the routing problem of this study, a state is defined by the current location of a vehicle and its destination and an action is defined as the next immediate destination node to be selected. For the reinforcement learning, a reward function is related to the goal of the problem. However because the objective of the problem is to minimize the travel time, a penalty function will be used instead of the reward function.

The travel time from a node to the next immediate node will be penalty of the corresponding state action pair. A value function specifies which decision is good in the long run, whereas a reward function indicates how good a decision is in the immediate future. For this study, the total travel time from a start node to a destination node will be the value function.

The following notation is introduced to describe the learning process.

t	:	The destination node of a current vehicle
k	:	The node where a current vehicle is located
(k,t)	:	The state of the vehicle, which consists of the current node *(k)* and the destination node *(t)*
A(k,t)	:	The set of candidates for the next node (action) from which a vehicle in state *(k, t)* may choose
a	:	An action taken by a vehicle, which is an element in A(k,t). The action corresponds to the next node selected.

g : The discount factor for future penalties ($0 \leq g \leq 1$). This study assumed *g =1, because the number of stages is finite in this study.*

r[(k,t),a] : The penalty, which is the travel time of a vehicle at state *s* from a current node to the next node (a). The travel time may also include the waiting time caused by traffic congestion.

Q[(k,t),a] : The expected discounted cumulative travel time, from the current node to the destination node, of a vehicle that select action *a*, at state (*k*, *t*).

The final output of the learning process is a decision matrix in which the numeric value in entry (k, t) represents the imminent next node of a vehicle at node k whose destination is node t. For obtaining the decision matrix, for a given state (*k*, *t*), the state-action pair with the lowest value of Q[*(k,t),a*] will be adopted as the value of entry (k,t) of the decision matrix. Thus, in order to obtain the decision matrix, it is only necessary to estimate the value of Q[(k,t),a]. The following equation is for updating the value of $\hat{Q}_n$ [(*k,t),a*] which is an estimator of Q[(*k,t),a*].

$$\hat{Q}_n[(k,t),a] = (1-a_n)\hat{Q}_{n-1}[(k,t),a] + a_n\{r[(k,t),a] + g\min_{a'}\hat{Q}_{n-1}[(k',t),a']\}, \qquad (1)$$

where $a_n = \dfrac{1}{1 + visits_n[(k,t),a]}$.

visits$_n$[(*k,t),a*] represents the total number of the visiting time during the learning process.
The conditional probability of selecting action *a*, given state[*k,t*], is calculated as follows:

$$p(a|(k,t)) = \frac{r^{\hat{Q}[(k,t),a]}}{\sum_{a\in A(k,t)} r^{Q[(k,t),a]-1}} \qquad (2)$$

where $a \in A(k,t)$ and r is a positive constant.

For implementing the learning process, a simulation model was developed by using eM-plant 7.6 versions. During the learning process, the action node is selected among the adjacent node set of current node of vehicle. In the initial state of the learning, $\hat{Q}$ [*(k,t),a*] is set to be the travel time from node k to t through node a under the assumption that the vehicle travels without any interruption by other vehicles during the travel.
However as experience accumulates, differences in Q[(*k*,*t*),*a*], among different next nodes for the current node and the destination node, become larger. For using the computational time efficiently, more effort must be devoted to estimating lower values of Q[(*k,t*),*a*] among different actions, which expression (2) attempts to do. According to (2), when Q[(k,t),a] for a next candidate node has a lower expected travel time among those of different candidate nodes at a given state, a higher probability of selecting the next candidate node results in a higher estimation accuracy of Q[(k,t),a]. As a result, more samples will be collected from traveling of vehicles from the current node to the candidate nest node.

4. TRAFFIC CONTROL RULES USED TO SUPPORT THE LEARNING PROCESS

During the learning process, it is necessary that traffic control rules must be provided for guaranteeing conflict and deadlock free travel of AGVs. In the following, several deadlock situations are described, which were found during the experiment and so some rules for resolving the deadlock situation must be provided. Figure 3 illustrates a possible deadlock situation, in which a cyclic deadlock may occur when requests of vehicles for the next nodes form a cycle of nodes. Figure 3 shows four vehicles from AGV1 to AGV4 and shaded nodes are the current locations of the vehicles. The arc represents the claim of a reservation for the next node for the travel by a vehicle on a node. For example, the vehicle on node 1 is claiming node 2 for the travel. However, if AGV1 is allowed to enter node 2, there will be a cyclic claim for the next node, which means the deadlock.

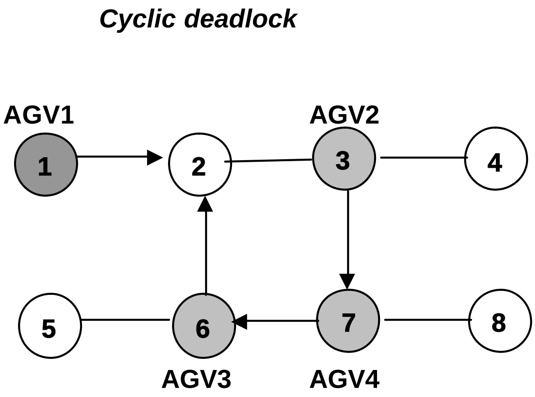

Fig 3. A cyclic deadlock situation on block lanes

There are two areas in which there are high possibilities of the deadlock situation. The first is the area of block lanes which consist of driving lanes of opposite directions or bidirectional lanes. The other area is between the lanes of the berth and those in front of blocks.
We use 'Semaphore' concept (Evers,1996) to prevent AGVS from cyclic deadlock situations. A semaphore is a classical solution to prevent resource deadlock. Whenever a vehicle arrives at semaphore area, the counting semaphore is triggered to check the availability of resources. The control logic of the counting semaphore can be defined by the following procedure of '*wait and proceed.*

Wait and proceed
When a vehicle predicts a deadlock on its route, the vehicle stops at its entry location and waits until at least one vehicle gets cleared from the predicted deadlock region.

Wait: if the capacity of semaphore is the same as the number of resources occupied, then wait until the occupied number of resources becomes smaller than the capacity.
Proceed: if the capacity of *semaphore* is greater than the number of occupied resources, then the number of occupied resources = the number of occupied resources + 1, and proceed to this semaphore area.

Semaphore areas with the capacity of 4 are illustrated in Fig 4. A semaphore, SP_i, is a set of nodes for which vehicles can request a reservation. In some cases, there may be nodes in an overlapped area by more than one semaphore areas. For example a vehicle arrives at node3 and the next visiting node is 4 triggers the counting semaphores, SP_1 and SP_2. The set of nodes in SP_1 is {1,2,3,4} and the set of nodes in SP_2 is {3,4,5,6}. Only when of the numbers of reserved resources in both sets are smaller than the capacities of the two semaphore areas, which are 3, then vehicle can enter the node and then the number of reserved resources is updated; otherwise the vehicle must wait until the conditions are satisfied.

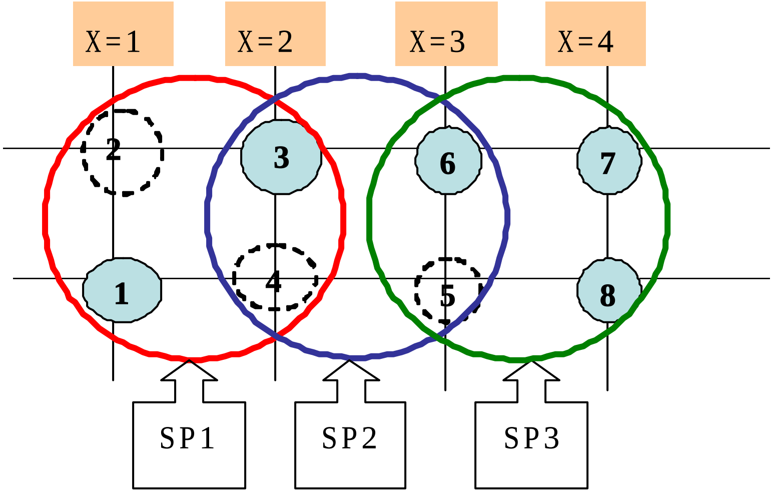

Fig 4. An example of semaphore areas in block lanes

Figure 5 illustrates a head-to-head conflict on a bidirectional path segment. In the simulation of this study, when a conflict of this type was detected, a detour, instead of the original route, was selected for avoiding this conflict.

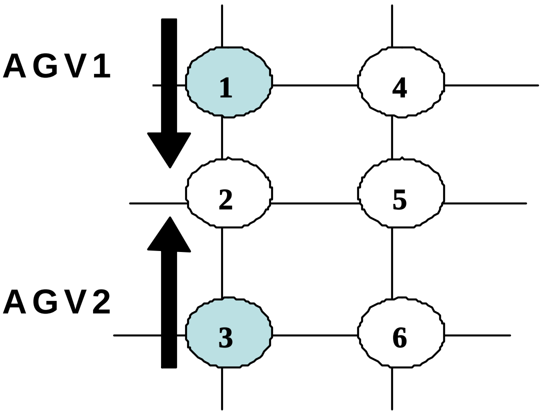

Fig 5. Head to Head conflict at bidirectional path

5. SIMULATION STUDY

5.1 Simulation scenario

A simulation program was developed by using the Em-plant simulation package for testing the routing method for vehicles. A container terminal with one berth of the length of 360m, 3 QCs, and 7 storage blocks, each of which has 3 transfer points, was modeled. The guide path network corresponded to that in Figure 3. The size of a node was assumed to be large enough to cover a vehicle. We assumed that the length of all the nodes was 16*m* and the total number of nodes in the terminal was 320. Five vehicles were assigned to each QC. During the simulation, each vehicle was dedicated to a single QC.

The simulation study was conducted for three different scenarios which have different delivery requirements as shown in Table 3. Because the delivery requirements are different from each other, the resulting route for the same starting node and the destination may be different in different scenarios.

The value of $\hat{Q}[(k,t),a]$ is updated by using expression (1), whenever a vehicle arrives at a node until it arrives at node t which is the destination. If the change of a $\hat{Q}$ is smaller than a pre-specified small value, ε, then the count of the stability is increased by one. If the change of a $\hat{Q}$ is greater than or equal to ε, then the count of the stability is reset to zero. When the count becomes to exceed a pre-specified limit, the learning process stopped. By using the final values of $\hat{Q}[(k,t),a]$, we can construct the final imminent destination matrix (decision matrix) by inserting a* = $\underset{a}{\operatorname{argmax}}\,\hat{Q}[(k, t), a]$ into entry (k,t). The decision matrix derived like this will be used for determining the travel route of a vehicle of each delivery request.

5.2 Simulation results

The simulation program was developed for learning and comparison purposes by using Em-plant simulation package which was run on the process of Pentium IV of 3Gh and memory of 1GB. For the simulation modelling, the configuration of the terminal in Fig 2 was assumed. Three scenarios of the delivery requirements as shown in Table 1 were assumed for the simulation.

The travel time of vehicles using routes obtained from the learning method was compared with that from the shortest distance routes which are being used in the most popular in practice. Table 2 shows the ratio of the travel time from the learning method to that from the shortest distance routes. The travel time consists of the moving time and the waiting time. It was found that the learning-based routing method outperformed the shortest travel distance route in the travel time. In the experiment, the value of ε was set to be 0.01. The counter of the stability was set to be 3000. Table 3 shows the computational time and the number of containers delivered for different values of ε.

Table1. An example of QC work schedule for simulation

Delivery demand	QC ID	Assigned block ID	Ratio of containers handled
1	1	1,2,3	39%
	2	3,4,5	36%
	3	5,6,7	25%
2	1	1,2,3	28%
	2	3,4,5	41%
	3	5,6,7	30%
3	1	1,2,3	27%
	2	3,4,5	34%

	3	5,6,7	38%

Table2. Comparison of travel time between the learning method and the shortest distance route (SDR)

Delivery demand	Q-learning process	SDR
1	94%	100%
2	96%	100%
3	80%	100%

Table3. Computational time for different values of e

Delivery demand	e range	Learning time (min)	Number of containers	Travel time
1	0.01	250	161,500	100%
	0.03	200	146,300	100.19%
	0.05	140	127,300	102.54%

6. CONCLUSION

This study applied a Q-learning algorithm to a routing planning for AGVs in port container terminals. The goal of routing planning is to find routes with the shortest traveling time for each delivery demand. It was shown how the Q-learning method can be used to estimate the expected travel time of vehicles between two nodes in the guide path network.

Through a simulation study, the performance of learning algorithm was compared with that of the shortest travel distance routes. It was shown that the travel time can be reduced by 10% by using the learning based routes instead of the distance based routes. For the future study, the results in this study should be generalized by much more extensive experiments. Moreover, the approach in this study can be extended to the problem of designing guide path networks for AGVs.

ACKNOWLEDGEMENTS

This study is partially supported by the Korean-German international symposium program of KOSEF in Korea and DFG in Germany and by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (The Regional Research Universities Program/Institute of Logistics Information Technology)

REFERENCES

Broadbent, A. J., Besant, C. B., Premi, S. K., and Walker,S.P.,1985, ”Free ranging AGV Systems: Promises, Problems and pathways,” *Proceeding of the 2nd international conference on automated materials handling*, pp.221-237.

Evers, J. J. M., Koppers.S. A. J., 1996,”Automated guided vehicle traffic control at a container terminal,” *Transport Research A*, 30(1), pp.21-34.

Gaskins, R. J., and Tanchoco, J. M. A., 1987,”Flow path design for automated guided vehicle systems”, International Journal of Production Research, 25(5), pp.667-676.

Kim, C. W., and Tanchoco, J. M. A., 1991, ”Conflict free shortest time bi-directional AGV routing”, *International Journal of Production Research*, 29(12),pp.2377-2391.

Lim, J. K., Lim, J. M., Yoshimoto, K., Kim, K. H., Takahashi,T., 2002,”A construction algorithm for designing guide path of automated guided vehicle system”, *International Journal of Production Research*, 40(15), pp.3981-3994.

Mahadevan, S.,1996, ”Average reward reinforcement learning; foundation, algorithms, and empirical results”, Machine Learning, 22(1), pp.159-195.

Mitchell, T .M., 1997, Machine learning, McGraw-hill, New York, pp.367-390.

Oboth, C., Batta, R., Karwan,M., 1999, ”Dynamic conflict free routing of automated guided vehicles”, *International Journal of Production Research*, 37(9), pp.2003-2030.

Rajotia, S., Shanker, K., Batra, J. L., 1998, ”A semi-dynamic time window constrained routing strategy in an AGV system”, *International Journal of Production Research*, 36(1), pp.35-50.