International Journal of Navigation and Port Research
Vol.30, No.9 pp. 773~777, 2006 (ISSN-1598-5725)

Routing and Collision Avoidance of Linear Motor based Transfer Systems using Online Dynamic Programming

†*Jeong-Tae Kim* · Hyun-Cheol Cho** · Kwon-Soon Lee****

*, **, ****Dept. of Electrical Eng., Dong-A University, 840 Hadan 2-dong, Saha-gu, Pusan 604-714*

Abstract : *Significant increase of container flows in the marine terminals requires more efficient port equipments such as logistic and transfer systems. This paper presents collision avoidance and routing approach based on dynamic programming (DP) algorithm for a linear motor based shuttle car which is considered as a new transfer system in the port terminals. Most of routing problems are focused on automatic guided vehicle (AGV) systems, but its solutions are hardly utilized for LM based shuttle cars since both are mechanically different. Our proposed DP is implemented for real-time searching of an optimal path for each shuttle car in the Agile port terminal located at California in USA.*

Key words : *Routing, Collision avoidance, Online dynamic programming, LMTT*

1. Introduction

Progressive increase of container flows in marine terminals requires efficient conveyance systems, such as unmanned container trailers, automation guided vehicles (AGVs), and linear motor (LM) based shuttle cars Arora et al.(2000). Especially, the latter is considerably focused on future transfer systems in marine applications because of its low maintenance cost and high reliability. However, its unique characteristics require highly complex routing with collision and deadlock avoidance for multiple shuttle cars.

Research for routing unmanned vehicles has provided interesting problems for engineering fields such as robotics, manufacturing, and port systems. In Desaulniers et al.(2003), the authors determined the shortest tour of a single free-ranging AGV, which is an incidence of the asymmetric traveling salesman problem, with a neural network approach. The authors in Dorigo et al.(1996) proposed a control strategy to guarantee no collision between unmanned vehicles at the junction points. In Ioannou et al.(2000), a routing algorithm for multi-mobile robots in transportation was explored with a modification of the "ant" optimization Ioannou et al.(2000), which is based on a colony of cooperating agents. Norman Konishi et al.(2002) suggested a recursive search algorithm to repeatedly evaluate each feasible route when a vehicle encounters a workstation. The authors in Norman(2002) presented a methodology for solving the simultaneous dispatching and conflict-free routing of AGVs in manufacturing systems. More recently, Petri-net based modeling was proposed to handle conflict and deadlock in AGV systems Soylu et al.(2000); Wu and Zhou(2004).

Most of the research in this area has to date focused on AGV systems and has not provided solutions for LM based shuttle cars. Because the two systems have significant mechanical differences, it is not possible to use AGV research to solve problems associated with LM based shuttle cars Arora et al.(2000). Thus, an innovative routing approach is required to operate LM systems with conflict and deadlock avoidance.

We present a simple network model for multiple LM based shuttle cars. We also propose their routing scheme using real-time dynamic programming (DP), which is an extended version of typical DP Winston(1994). Although currently few container terminals use LM, their use is expected to grow in the future because of their known advantages. Automatic container terminals using LM are known as an Agile port Yoo et al.(2005) and can be designed to handle 2,482,000 TEU per year and serve one ship every 24 hours.

The remainder of this paper is as follows: Section 2 provides preliminary background for dynamic programming algorithm. In Section 3, we present network model for a container terminal with multiple shuttle cars. In Section 4, we propose our online DP routing algorithm. Simulation example and conclusion are respectively provided in Section 5 and 6.

---

† Corresponding Author : Jeong-Tae Kim, tobeluxunybrain@hanmail.net, 016-9844-0112
** hyuncchо@gmail.com, 051)200-6950
*** kslee@dau.ac.kr

2. Dynamic programming

DP is popular in applications of routing problem, which is based on backward computation so that its computational time is efficiently low. This method is accomplished in sequence of decision. We first describe how the DP algorithm is worked for routing application.

1) Time assumption is calculated through all distances among nodes from destination to directly connected nodes and these nodes is checked.
2) Time assumption for distances among checked nodes. We already obtain distances from destination to checked nodes knowing its values. Thus, we easily have a minimum distance value if using a minimum value from previous stage.
3) We calculate distance repeatedly in the same way at the next stage until a final stage.

This is obviously allows that DP is usually suitable for our routing problem including collision and deadlock avoidances as well as much more quick computational speed in real-time implementation.

3. Modeling of a container yard

LM based shuttle cars move along a monorail in the container terminal to convey container boxes among several workstations. We model the container yard in Yoo et al.(2005) for the cars with a mesh network topology (see Fig. 1). This model does not include gates through which containers flow, ships traffic in and out of the yard, and buffers for gates and trains. We modify the terminal model only to represent paths and workstations for shuttle cars. In Fig. 2, nodes n_{ij}, $i = 1, \cdots, M$, $j = 1, \cdots, N$, indicate workstations where a crane loads or unloads container boxes to shuttle cars. Links between two nodes represent a feasible path for a shuttle car. Each link has weights $w_{ij} > 0$, $i = 1, \cdots, M$, $j = 1, \cdots, N-1$ and $v_{ij} > 0$, $i = 1, \cdots, M-1$, $j = 1, \cdots, N$, which represent the expected transit time between two nodes for a shuttle car. These weights are regarded as costs in our routing problem. For simplicity, we ignore service time at the workstations and other costs associated with transit between workstations.

In Fig. 1, the network is apparently formed with a matrix in which its link distance is equal in each node. We consider this characteristics in this yard model. A LMTT shuttle car is different from a AGV which has independent process to find an optimal path with collision avoidance. As well, an AGV flexibly drives curved loads changing directions. But a LM based shuttle car must stop to switch its direction. This main difference between both is significantly considered in control applications as well as routing problem for multiple shuttle cars.

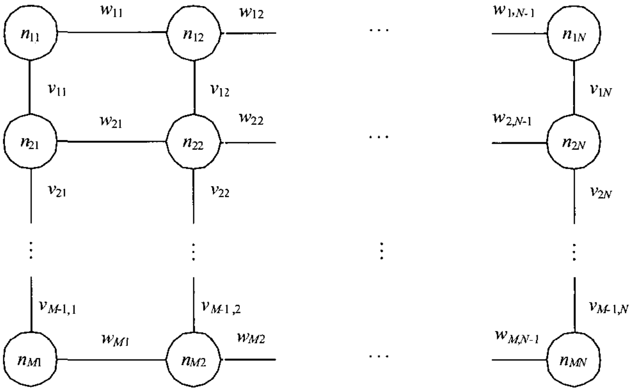

Fig. 1 Modeling of container yard.

4. Real-time DP algorithm for routing of multiple shuttle cars

Routing of shuttle cars in Fig. 1 involves finding the shortest path given a starting node $s \in n_{ij}$ and a destination node $d \in n_{ij}$ where $s \neq d$. For a single shuttle car in the network of Fig. 1, its solution is easily obtained using a typical DP algorithm. However, routing for multiple cars raises possibility of collision. This is simply defined as two or more cars simultaneously occupying the same segment or workstation within the same time interval. Conflict-free routing is possible by selecting a suboptimal path for cars with lower priority to avoid collision.

We propose a real-time DP algorithm for routing of multiple shuttle cars with collision avoidance. First, we determine optimal paths for all of shuttle cars by means of a typical DP algorithm under each starting node and destination without delay time for changing its direction. Next, we apply delay time at the corner of the network which is more realistic consideration in real-time implementation. In this scenario, number of possible optimal paths is decreased than the first case. We check its optimal paths whether any identical path in same time interval is occurred. If no collision path is expected, determined optimal paths are used, otherwise, the paths are recalculated in considering priority. In other words, route of a car with higher priority is kept without any modification, but its path for a lower-priority car is recalculated reruning DP algorithm, thus alternative paths are sought for cars with lower priority. In practice, the priority is determined from job scheduling, but we do not deal with it in this paper. A

DP is iteratively run until we obtain a satisfactory path excluding the collision node for the cars with lower priority. But there are some burst situation while LMTT car is running, for instance, rail and shuttle car are broken, time what to arrive node in route happen difference. The proposed algorithm is schematically summarized in Fig. 2.

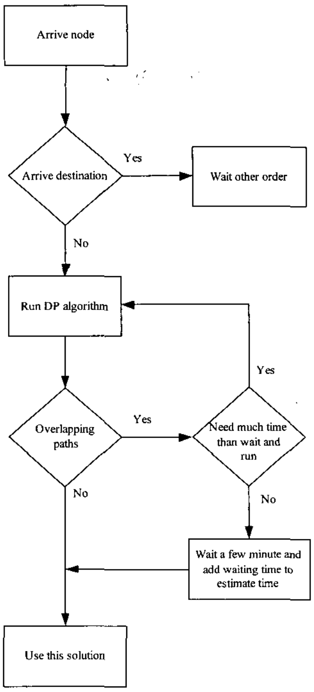

Fig. 2 Flowchart of online DP algorithm for multiple shuttle cars

5. Simulation Example

In this Section, we evaluate our routing method applying to Agile terminal model. We consider the mesh network with four vertical and five horizontal nodes, shown in Fig. 3.

Case I : In the mesh model, we identically define the weights for simplicity. We assume that there are three cars in the network with starting nodes, s_1=2, s_2=20, s_3=17 and destinations d_1=20, d_2=6, d_3=5, and the third car starts later than the others. Their routes using typical DP algorithm for this simulation scenario is shown Fig. 4. From these curves, we realize all shuttle cars have two optimal paths and expect collisions of car 1 and 3 in node 12 after 6 minutes, car 1 and 2 in node 17 after 9 minutes, and car 1 and 3 in node 15 after 17 minutes. We find a collision-free optimal route for each car based on this result (see Table 1).

Next, we assume car 2 is delayed to reach node 10 starting from node 20 due to unexpected disturbance, thus arrived after 11 minutes (9 minutes expected originally). This unexpected situation generates the determined paths for car 2 are suboptimal such that we rerun DP algorithm when car 2 arrives at node 15. From new optimal path of car 2, we find collision of car 1 and 2 at node 10 in 14 minutes (see Fig. 5). Avoiding deadlock and fixing the path for car 2, we finally determine the optimal path for both (see Table 2). Time-history of this new path is shown in Fig. 6.

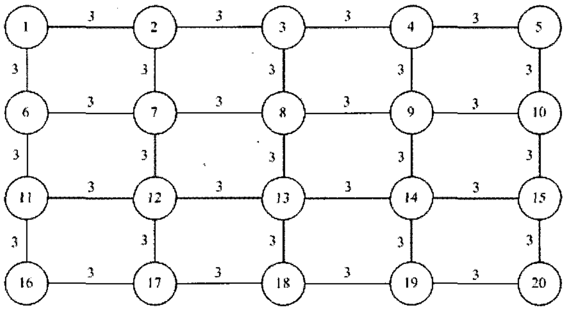

Fig. 3 Network with 4 by 5 mesh (*Case* I).

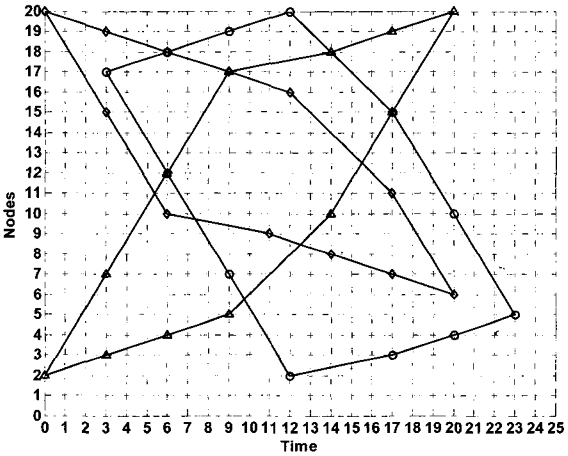

Fig. 4 Time histories of each car by typical DP (*Case* I) (△:car 1, ◇:car 2, ○:car 3).

Table 1 Optimal paths for each car by typical DP (*Case* I).

Car No.	Optimal path	Path Duration
Car I	2-3-4-5-10-15-20	20 min
Car II	20-15-10-9-8-7-6	20 min
Car III	17-12-7-2-3-4-5	23 min

Table 2 Optimal paths for each car by online DP (*Case* I).

Car No.	Optimal path	Path Duration
Car I	2-3-4-5-10-15-20	20 min
Car II	20-15-14-13-12-11-6	30 min
Car III	17-12-7-2-3-4-5	23 min

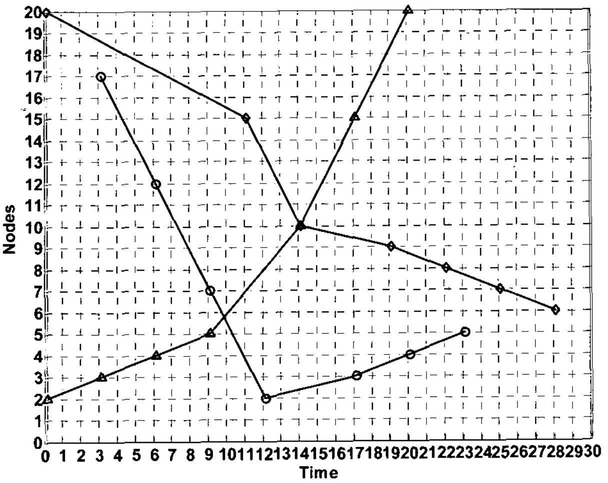

Fig. 5 Time histories of each car by online DP (*Case* Ⅰ)
(△:car 1, ◇:car 2, ○:car 3).

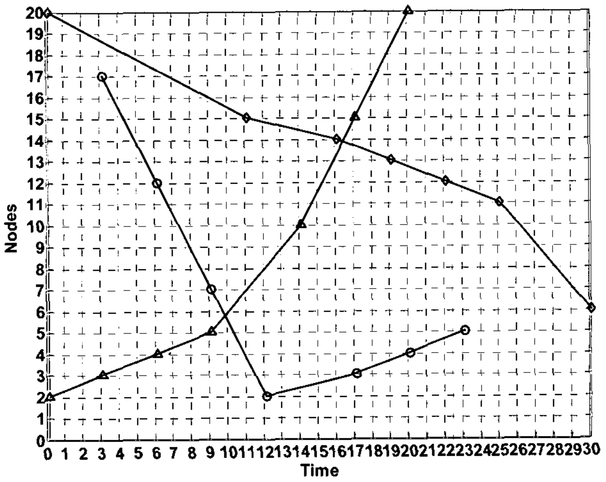

Fig. 6 Time histories of each car by online DP (*Case* Ⅰ)
(△:car 1, ◇:car 2, ○:car 3).

Case Ⅱ : We define different weights in the network shown in Fig. 7. Start and destination nodes of the three cars are s_1=3, s_2=18, s_3=15 and d_1=20, d_2=5, d_3=1, and consuming time for any corner is 2 min. After running the DP algorithm for this simulation scenario, we obtain optimal paths for these cars. Fig. 8 illustrates time histories of their routes and Table 3 provides time consumptions for these routes. We observe from this result that car 1 and car 3 are expected to be collision in node 9 and 10 after 4 min. Car 3 have a suboptimal path rerunning the DP algorithm to solve this problem. We have final optimal routes for these cars without any collision and dead lock. Fig. 9 and Table 4 shows the optimal paths and total time consumptions, respectively. We realize total time consumption of car 3 is increase by 1 min.

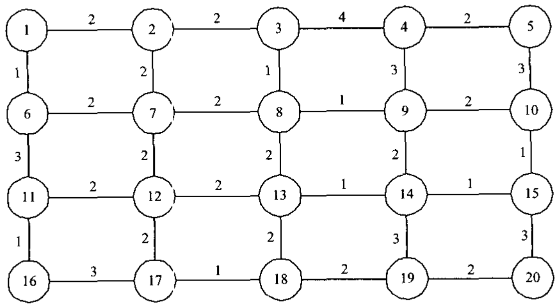

Fig. 7 Network with 4 by 5 mesh(*Case* Ⅱ).

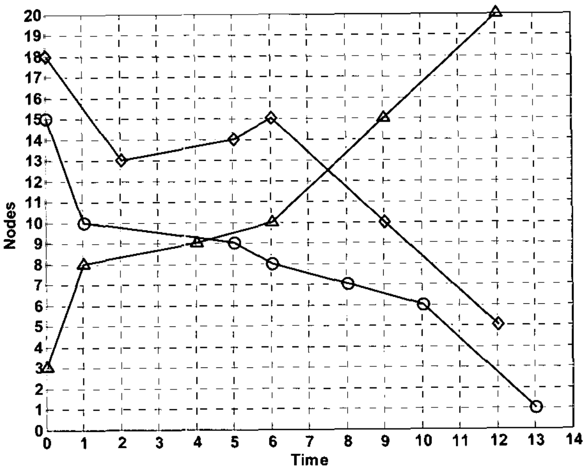

Fig. 8 Time histories of each car by typical DP(*Case* Ⅱ)
(△:car 1, ◇:car 2, ○:car 3).

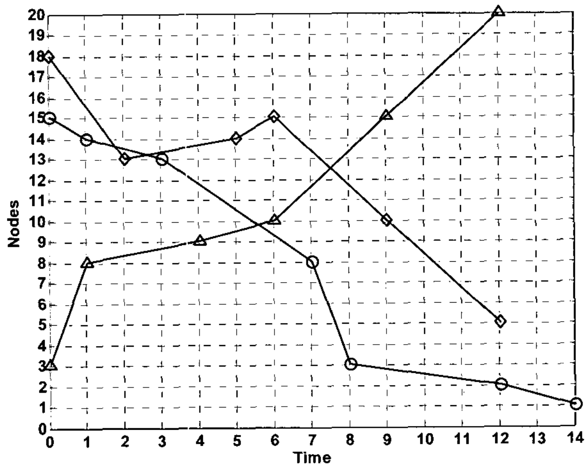

Fig. 9 Time histories of each car by online DP (*Case* Ⅱ)
(△:car 1, ◇:car 2, ○:car 3).

Table 3 Optimal paths for each car by typical DP(*Case* Ⅱ).

Car No.	Optimal path	Path Duration
Car Ⅰ	3-8-9-10-15-20	12 min
Car Ⅱ	18-13-14-15-10-5	12 min
Car Ⅲ	15-10-9-8-7-6-1	13 min

Table 4 Optimal paths for each car by online DP (*Case* II).

Car No.	Optimal path	Path Duration
Car I	3-8-9-10-15-20	12 min
Car II	18-13-14-15-10-5	12 min
Car III	15-14-13-8-3-2-1	14 min

6. Conclusion

We present a network model for LM based multiple cars in marine terminals and propose a novel algorithm for their collision free routing. A DP algorithm is iteratively run until we obtain suboptimal paths that eliminate the collision associated with the optimal DP solution. We demonstrate the results using simple simulation scenarios which have several operational constraints. Future work will include modeling and optimal routing for a more complex port system represented by a more complex network topology in which we consider realistic materials in port terminals. We will also apply queuing theory and consider the effect of service time on the model.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M10300000030306J000030310)

References

[1] Arora, S., Raina, A., and Mittal, A.(2000), "Collision avoidance among AGVs at junction," *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 585-589.

[2] Desaulniers, G., Langevin, A., Riopel, D., and Villeneuve, B.(2003), "Dispatching and conflict-free routing of automated guided vehicles: An exact approach," *The Int. Journal of Flexible Manufacturing systems*, vol. 15, pp. 309-331.

[3] Dorigo, M., Maniezzo, V. and Colorni, A.(1996), "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29-41.

[4] Ioannou, P., Kosmatopoulos, E., Jula, H., Collinge, A., Liu, C. and Asef-Vaziri, A.(2000), "Cargo handling technologies," *Technical Report*, Univ. of Southern California.

[5] Ioannou, P., Jula, H., Liu, C., Vukadinovic, K., and Pourmohammadi, H.(2000), "Advanced material handling: automated guided vehicles in Agile ports," *Technical Report, Center for Advanced Transportation Technologies*, University of Southern California.

[6] Konishi, M., Nishi, T., Nakano, K., and Sotobayashi, K.(2002), "Evolutionary routing method for multi mobile robots in transportation, *Proc. of the IEEE Int. Symposium on Intelligent Control*, pp. 490-495.

[7] Norman, M.(2002), "A recursive method for traffic management through a complex path network," *Proc. of the Simulation Conf.*, pp. 537-541.

[8] Soylu, M., Ozdemirel, N. and Kayaligil, S.(2000), "A self-organazing neural network approach for the single AGV routing problem," *European Journal of Operational Research*, vol. 121, pp. 124 - 137.

[9] Wu, N. and Zhou, M.(2004), "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. On Mechatronics*, vol. 9, no. 1, pp. 50-87.

[10] Winston, W.(1994), "*Operations research*, Duxbury", Belmont.

[11] Yoo, J. W., Sim, E. S., Cao, C. and Park, J. W.(2005), "An algorithm for deadlock avoidance in an AGV system," *Int. J. Adv. Manuf. Technol.*, vol. 26, pp. 659-668.

Received 22 November 2006
Accepted 9 January 2007