한국항해항만학회 제30권 제2호 추계학술대회논문집 pp. 393~397, 2006. 12

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 Engineering, Dong-A University, Pusan 604-714, Korea

* tobeluxurybrain@hanmail.net, ** hyunccho@gmail.com, *** kslee@dau,ac,kr

ABSTRACT : Significant increase of container flows in marine terminals requires more efficient automatic port systems. This paper presents a novel routing and collision avoidance algorithm of linear motor based shuttle cars using dynamic programming (DP). The proposed DP is accomplished online for determining optimal paths for each shuttle car. We apply our algorithm to Agile port terminal 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 [1]. 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 [2], 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 [3] proposed a control strategy to guarantee no collision between unmanned vehicles at the junction points. In [4], a routing algorithm for multi-mobile robots in transportation was explored with a modification of the "ant" optimization [5], which is based on a colony of cooperating agents. Norman [6] suggested a recursive search algorithm to repeatedly evaluate each feasible route when a vehicle encounters a workstation. The authors in [7] 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 [8], [9].

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 [1]. 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 a novel scheme for their routing using online dynamic programming (DP), which is an extended version of typical DP [10]. 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 [11], 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.

2. Background

Dijkstra algorithm and dynamic programming are popular in applications of routing problem. However, the former has much more time assumption for determining of an optimal path since all nodes are involved for computation. The latter is related to backward computation, but its computational time is obviously lower. Thus, the DP is usually suitable for our routing problem including collision and deadlock avoidances.

Dynamic Programming

Dynamic programming is accomplished in sequence of decision. We first describe how the DP algorithm is worked for routing application.

a) Time assumption is calculated through all distances among nodes from destination to directly connected nodes and these nodes is checked.
b) 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.
c) We calculate distance repeatedly in the same way at the next stage until a final stage.

Example: Consider the case in Figure 1 with a starting node S and a final node T. First, we know nodes C_2, C_3, and C_4 are closed to a node T and thus obtain its distances between them respectively, as $C_2 = 2$, $C_3 = 4$, $C_4 = 5$. Here, C_1 is nearby C_2 and C_3 that are already checked and obtain distance from C_2 and C_3 to C_1. Thus, node C_1 has two distance value as $C_1 = 5$ (from C_2) and 6 (from C_3) and we have the shortest path from C_1 to T as $C_1 \rightarrow C_2 \rightarrow T$ with 5 total distances. Next, S is neighbored to C_1 and C_4 which are checked nodes in the previous step, thus we obtain distances between these nodes as $S = 7$ (from C_1) and 8 (from C_4). We reach current node S to calculate a minimum value for a shortest path. Finally, we find a optimal path is $S \rightarrow C_1 \rightarrow C_2 \rightarrow T$ and its total distance is 7. ■

DP algorithm is somewhat different than other searching algorithms, which is bottom-up computational strategy. This allows much more quick computational speed in online

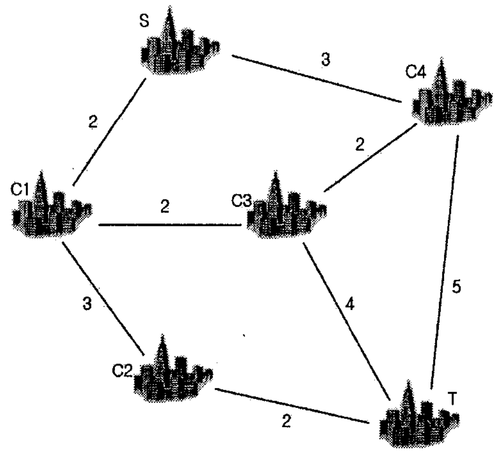

Figure 1. A network with 6 nodes.

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 [11] for the cars with a mesh network topology (see Figure 2). 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 Figure 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 $w_{ij} > 0$, $i = 1, \cdots, M$, $j = 1, \cdots, N\text{-}1$ and $v_{ij} > 0$, $i = 1, \cdots, M\text{-}1$, $j = 1, \cdots, N$, 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 Figure 2, 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.

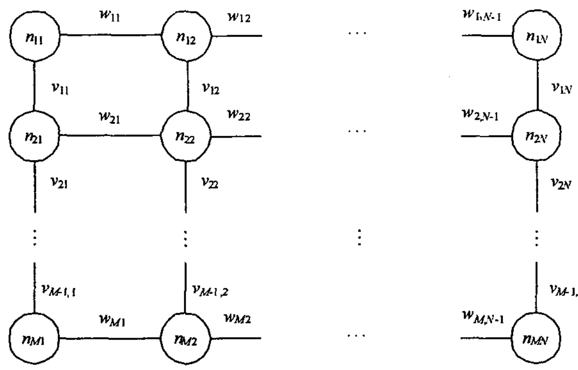

Figure 2. Modeling of container yard.

4. Online DP algorithm for routing of multiple shuttle cars

Routing of the shuttle cars in Figure 2 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 Figure 2, the solution is easily obtained using a typical DP algorithm. However, routing for multiple cars raises the 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 online 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 change direction. Maybe there are many optimal paths what determined by typical DP. Next, we apply delay time at the feasible optimal paths. This calculation make decreasing number of optimal paths. Then, we compare optimal paths to detect any identical paths in same time interval, which would result in a collision. If no collision node is expected, the optimal paths are directly applied for each car as its route. Otherwise, the optimal path is only used for the highest priority car and alternative paths are sought for cars with lower priority. A DP is iteratively run until we obtain a satisfactory path excluding the collision node for the cars with lower priority. Then, optimal paths are applied for each car as its route. 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. Therefore everytime DP is run when to arrive node in route and compare with path of the other cars. But when another path need more time than the other order, the other word, wait and run, that path is unnecessary. The proposed algorithm is schematically summarized in Figure 3.

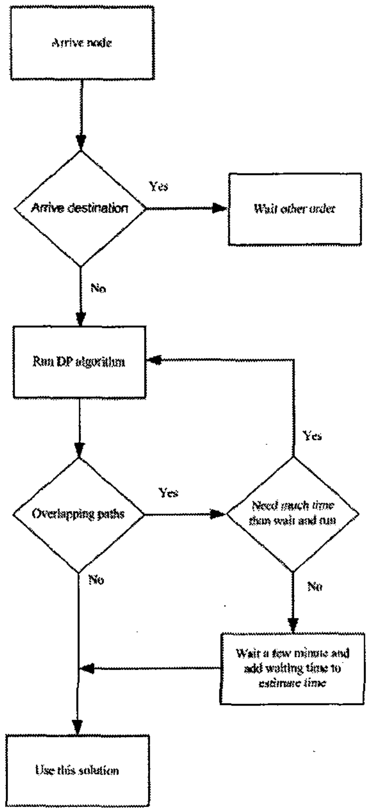

Figure 3. Flowchart of online DP algorithm for multiple shuttle cars.

5. Simulation Example

Consider the mesh network with four vertical and five horizontal nodes, shown in Figure 4. We identically let weights between each node three in which there are many opportunities for collision of shuttle cars. 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 Figure 5. 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 online 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. 6). Avoiding deadlock and fixing the path for car 2, we finally determine the optimal path for both (see Table 2).

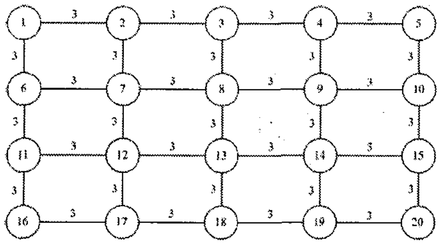

Figure 4. Network with 4 by 5 mesh.

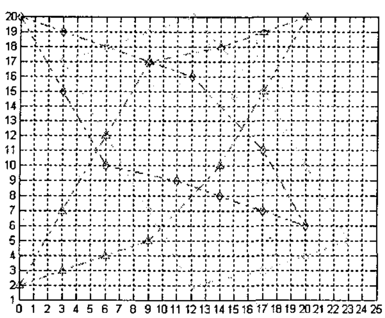

Figure 5. Time histories of each shuttle car by typical DP (△:car 1, ◇:car 2, ∘:car 3).

Table 1. Optimal paths for each car by typical DP.

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

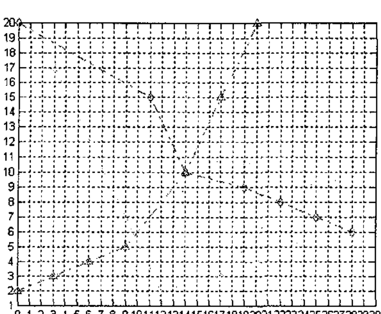

Figure 6. Time histories of each shuttle car by online DP (△:car 1, ◇:car 2, ∘:car 3).

Table 2. Optimal paths for each car by online DP.

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

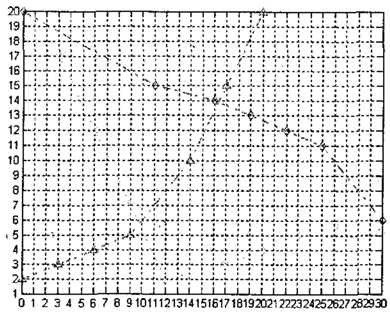

Figure 7. Time histories of each shuttle car by online DP (△:car 1, ◇:car 2, ∘:car 3).

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. We will also apply queuing theory and consider the effect of service time on the model.

Acknowledgement

This work was supported by the National Research Laboratory (NRL) program of the Korean Ministry of Science and Technology (MOST).

References

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

[2] M. Soylu, N. Ozdemirel, and S. Kayaligil, "A self-organazing neural network approach for the single

AGV routing problem," *European Journal of Operational Research*, vol. 121, pp. 124 - 137, 2000.

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

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

[5] M. Dorigo, V. Maniezzo, and A. Colorni, "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, 1996.

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

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

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

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

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

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