Time Window Based Genetic Algorithm for Multi-AGVs Conflict-free Path Planning in Automated Container Terminals

Tong Lu[1], Zhao-Hui Sun[1,*], Siqi Qiu[1], Xinguo Ming[1]
[1]Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai, China
{Emma_lu, zh.sun, siqiqiu, xgming}@sjtu.edu.cn

2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) | 978-1-6654-3771-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEEM50564.2021.9672805

Abstract - **The continuously increasing demand for containers makes it difficult for traditional ports to bear the heavy workload. As a result, the transformation of traditional ports into automated container terminals (ACTs) has already become a vital trend for the shipping industry. Nowadays, the automated guided vehicle (AGV) is widely used in ACTs. It becomes one of the main handling equipment for the loading and unloading of goods. However, the conflicts that occur among AGVs will influence the operating efficiency of ACTs. To achieve conflict-free path planning, the problem of path planning is modeled as min-max mathematical programming in this paper. Then, the time window-based Genetic Algorithm (TWGA) is proposed to solve the problem. To verify the performance of TWGA, different cases with small-scale and large-scale are used to test the ability to find the optimal solution of the algorithm. Results illustrate that the proposed algorithm could effectively reduce the conflicts among AGVs, and outperforms the compared algorithm.**

Keywords - **Multi-AGVs path planning, Conflict-free, Time window, Genetic algorithm.**

I. INTRODUCTION

The boom in global trade has been pushing up the development of automated container terminals (ACTs), where more and more intelligent logistics technologies and equipment have been used. Among them, Automated guided vehicle (AGV) plays a vital role in significantly reducing transportation costs and improving transportation efficiency [1]. This makes the path planning of AGVs becoming one of the significant concerns in the logistics domain.

Path planning is a crucial aspect of automatic container terminal design. It affects the time necessary for horizontal transport, the scheduling of AGVs, and the overall efficiency of the system [2]. Various methods have been proposed to solve this problem. Wang et al. [3] introduced factors of turning, and edge removal based on the improved A* algorithm to solve the *k* shortest path problem and proposed a dynamic path planning method based on the A* algorithm to search the shortest-time path and avoid collision effectively. Guo et al. [4] adopted an improved Dijkstra algorithm that can find all equidistant shortest paths to solve the path-planning problem in the rectangular environment. Zhong et al. [5] proposed a priority-based speed control strategy in conjunction with the Dijkstra depth-first search algorithm to solve the AGV path planning problem. Fan et al. [6] proposed a searching strategy F-TWS (Forward Time Window Searching) to search for passable time windows along candidate paths for AGVs. However, these algorithms search for the solution by setting some harsh conditions, which speeds up the searching process but affects the accuracy meanwhile. Hence, in this paper, a time window-based genetic algorithm (TWGA) based on genetic algorithm (GA) is proposed, which searched for the optimal solution in a relatively small range to obtain a better solution than the traditional methods like A*, etc. At the same time, since the search scope is narrowed, the calculation time will not be very long.

The contribution of this paper is two-fold. First, we formulate the problem into a min-max mathematical programming model. The model takes advantage of the first come first served (FCFS) policy to avoid the conflict between AGVs when AGVs tend to occupy the same node at the same time. Second, TWGA is proposed to solve the problem. In TWGA, the time window is used to achieve conflict-free path planning. The combination of conflict-free path planning and GA ensures the ability to search for the optimal solution of the algorithm. Our experiment shows that the proposed TWGA outperforms the compared algorithm in different cases with small-scale and large-scale.

The remainder of this paper is organized as follows. Section II describes the problem based on a directed graph. Section III gives a detailed design scheme of TWGA. Section IV shows the results of the proposed algorithm and compares it with the F-TWS. Finally, the conclusion is discussed in Section V.

II. PROBLEM DESCRIPTION

A. Modeling of Layout in ACT into A Directed Graph

Fig. 1 shows the layout of an ACT. Quay cranes (QCs), yard cranes (YCs), and AGVs are handling equipment. The QCs are responsible for placing the container on the AGV from the ship or removing the container from the AGV [2]. The YCs are responsible for containers' interaction between the horizontal transportation area and the yard area. AGVs only run in the horizontal transportation area for removing containers. As shown in Fig. 1, each road has a prescribed driving direction. AGVs are not allowed to run against the prescribed direction.

978-1-6654-3771-4/21/$31.00 ©2021 IEEE

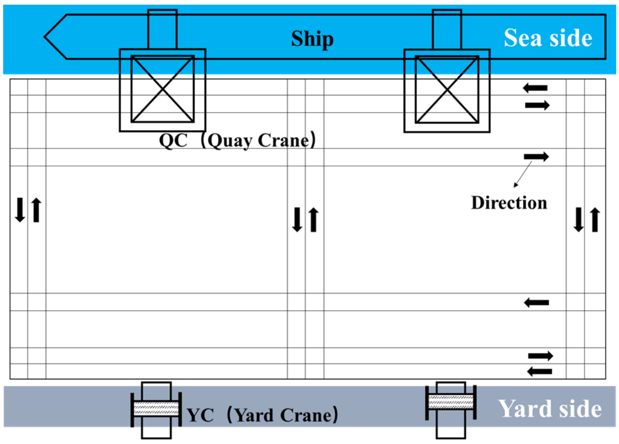

Fig. 1. The layout of ACT.

To formulate the problem, the layout is modeled into a directed graph. As shown in Fig.2, there are 16 nodes in total. Blue nodes are the QC nodes where containers are removed from the AGV to the ship or from the ship to the AGV. Similarly, red nodes are block nodes where containers are transferred. Red and blue blocks represent AGV loading/unloading ports that allow multiple vehicles to stop and run across.

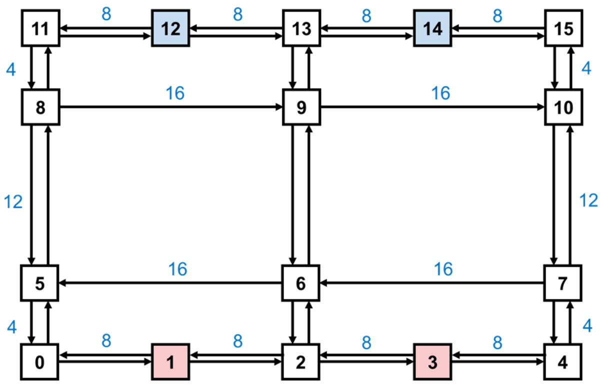

Fig. 2. The directed graph.

B. Problem Description

Given the directed graph of the ACT, the goal is to find a combination of paths that enable each vehicle to travel from its start node to the end node without conflicting with other AGVs, and at the same time, all the AGVs can finish the task as fast as possible. Here are the assumptions.

- The speed of AGVs is uniform. Therefore, the length of the path between two nodes can be represented by the number of unit times (numbers in blue next to the lines in Fig.2) that the AGV travels between the two nodes.
- Each AGV can serve any QC or YC.
- Each node can only be occupied by one AGV at the same time.
- The influence brought by the different ways of turning is ignored.
- The loading and unloading times of containers are 0.

1) Notation and definition

Notation	Definition
V	Set of AGVs, $V = (1,2,\dots,k)$
N	Set of all nodes
G	The directed line segment between nodes. The directed line segment is denoted by (i,j), which represents that the road is from node i to node j.
S	Set of start nodes for AGVs, $S = (s_1, s_2, \cdots, s_k)$
E	Set of end nodes for AGVs, $E = (e_1, e_2, \cdots, e_k)$
M	A large positive number
A	The adjacency matrix of the directed graph, $a_{ij} = 0\ or\ 1$, $(i,j) \in G$ is equivalent to $a_{ij} = 1$
x_{ij}^k	0,1 variable. $x_{ij}^k = 1$, if AGV k goes through road (i,j); $x_{ij}^k = 0$, otherwise
y_i^k	0,1 variable. $y_i^k = 1$, if AGV k waits at node i; $y_i^k = 0$, otherwise
$p_{k'k}^i$	0,1 variable. $p_{k'k}^i = 1$, if AGV k' and AGV k pass through the node i successively; $p_{k'k}^i = 0$, otherwise
tw_i^k	Waiting time of AGV k at node i
ta_i^k	The time that AGV k arriving at node i
tl_i^k	The time that AGV k leaving node i
l_{ij}	Length of the road (i,j)

2) Mathematical modeling

Before path planning, each vehicle is assigned a task, which is known. For example, for AGV k, it should run from the start node s_k to its end node e_k. CT_k represents the time AGV k completes its task and CT_{max} represents the maximum completion time for AGVs. They are defined by Eq. 1 and Eq. 2.

$$CT_k = \sum_{i \in N} l_{ij} x_{ij}^k + \sum_{(i,j) \in G} y_i^k tw_i^k \tag{1}$$

$$CT_{max} = \max_{k \in K}\{CT_k\} \tag{2}$$

To avoid the conflict problem of AGVs, the principle of first come first served (FCFS) is adopted [5]. The mathematical model can be formulated as follows:

$$f = Min\ CT_{max} \tag{3}$$

$$x_{ij}^k - x_{ji}^k = \begin{cases} 1, AGV\ k\ run\ through\ road\ (i,j) \\ -1, GV\ k\ run\ through\ road\ (j,i) \\ 0, otherwise \end{cases} \forall k \in K \tag{4}$$

$$ta_{s_k}^k = 0 \qquad \forall k \in K \tag{5}$$

$$ta_{e_k}^k = CT_k \quad \forall k \in K \tag{6}$$

$$ta_i^k + y_i^k tw_i^k \le tl_i^k \quad \forall i \in N, \forall k \in K \tag{7}$$

$$tl_i^k + l_{ij} - M(1 - x_{ij}^k) \le ta_j^k \qquad \forall (i,j) \in G, \forall k \in K \tag{8}$$

$$tl_i^{k'} - ta_i^k - M\left(1 - p_{k'k}^i\right) < 0 \quad \forall i \in N, \forall k', k \in K$$
(9)

$$\sum x_{ij}^k - M a_{ij} < 0 \quad \forall i, j \in N, \forall k \in K \qquad (10)$$

Eq. 3 is the objective function that defines the goal to minimize the maximum completion time of all AGVs. Constraint 4 is the direction constraint. Constraints 5-10 are constraints related to the AGV operation sequence and time. Constraints 5 and 6 ensure that AGV k runs from the start node s_k to the end node e_k. Constraint 7 ensures that the time AGV k leaves one node is no less than the time AGV k arrives at this node plus the time AGV k waiting at the current node. Constraint 8 means that if AGV k runs through the road (i, j), the time it enters node j can not be less than the time it leaves node i plus the time it's been on the road. Constraint 9 shows the FCFS principle. If two AGVs go through the same node, one cannot enter the node unless the other one has left. Constraint 10 means that AGVs only travel on the connected road.

III. ALGORITHM DESIGN

GA is a parallel and global search technique that emulates natural genetic operators. Due to many points in the parameter space are evaluated simultaneously, it is more likely to converge to the global optimal [7]. Therefore, it has been used in many fields, such as manufacturing [8-11]. The proposed TWGA to solve the model in section II is based on GA. The procedure of TWGA is shown in Fig. 3.

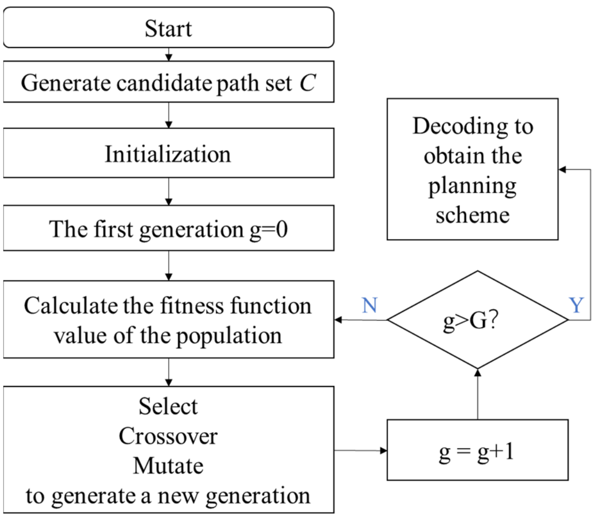

Fig. 3. The procedure of TWGA

A. Generate the candidate path set

The directed graph shown in Fig.2 means that, for each AGV with a known starting point and ending point, the shortest path can be easily calculated. In this paper, to reduce the computation time of TWGA, we narrow down the search scope of GA to a candidate path set *C*. *C* is composed of the candidate path set C_k of every AGV k, which consists of the shortest paths and the second shortest paths of AGV k.

$$C = \{C_1, C_2, \ldots, C_k\} \qquad (11)$$

The preparation of the whole algorithm is to generate the candidate path set *C*. Breadth-first search (BFS) [12] is used to generate *C*. Since the process of BFS is not the key point of the algorithm, we will not describe it in detail in this paper.

B. Initialization

The main purpose of initialization is to generate the first population. Before that, the *chromosome* should be encoded in a way that could simplify the calculation process. In this paper, the encoding method of the *chromosome* is shown in Fig.4. The length of the *chromosome* equals the number of the AGVs, so if the total number of AGVs is k, every *chromosome* comprises k *genes*. Every *gene* consists of two parts. As shown in Fig.4, the first part represents the ID of AGV, and the second part shows the selected path from the candidate path set C_2.

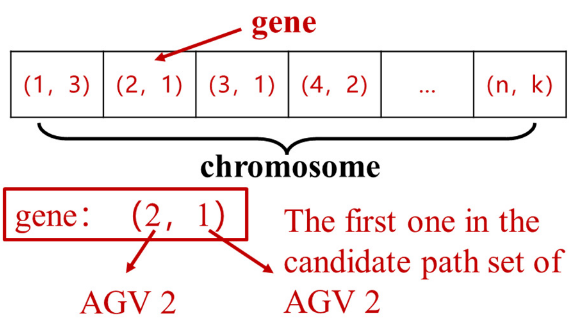

Fig. 4. Encoding of chromosome

C. Operations in GA

1) Selection

In this paper, individuals are select according to their fitness values. The retain rate is set to 40%, individuals with fitness values in the top 40% are selected as the parents of the next population, as long as the randomly generated number is less than the random select rate of 0.2.

2) Crossover

Randomly select two hybridization points from each parent, and then the hybridization fragments are exchanged between the two points. It should be noted that the crossover does not change the first part of the *gene*, but only the second part which represents the chosen candidate path. Otherwise, it may lead to the problem of missing an AGV or duplication of an AGV in the *chromosome*.

3) Mutation

The mutation is a means to realize population diversity and also a guarantee for global optimization. The specific design is as follows. According to the given mutation rate of 0.8, two integers are randomly selected for the selected mutant individuals to meet the requirements of $1<u<v<k$, the *gene* segment between u and v (including u and v) is mutated by replacing all candidate path numbers in the *gene* segment.

4) Fitness value calculation

Fitness value calculation is the conflict-free path planning of AGVs based on the selected candidate path of each AGV. According to the gene sequence on the *chromosome*, the candidate path of the AGV is planned successively. After the planning of one AGV is completed, the time window of all nodes on the directed graph should be updated before the next AGV is scheduled. The specific planning procedure is shown in Fig.5.

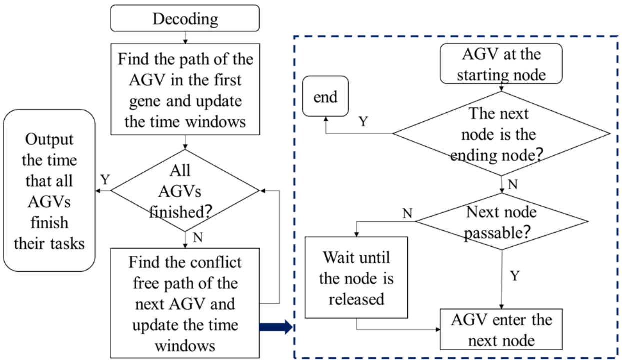

Fig. 5. The procedure of conflict-free path planning

To avoid collisions that multiple AGVs want to travel into the same block at the same time, the path planned for each AGV involves not only the nodes to move along but the corresponding passable time slot of each node in the path [6]. Therefore, whenever AGV moves from the current node to the next one, whether the next node is passable needs to be judged. If the answer is yes, then AGV moves to the next node successfully. If not, as shown in Fig.6, AGV has to wait at the current node *n'*, until the next node *n* is released.

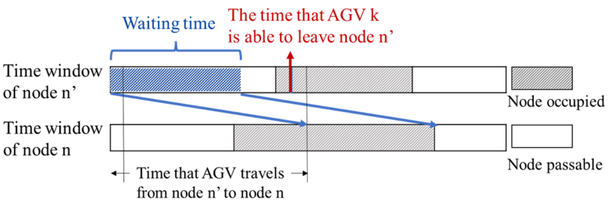

Fig. 6. AGV waits at the current node

However, if the situation in Fig.7 occurs, the path planning fails.

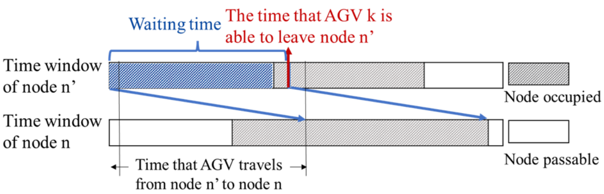

Fig. 7. Path planning fails

IV. EXPERIMENT

The experiments are run on Python 3.7 platform on a PC with Intel Core i7-8550U CPU/8 GB RAM.

A. Experiment setting

The scale of the AGV running site is a directed graph with 16 nodes shown in Fig.2. The number of AGVs ranges from 20 to 160, covering both small-scale and large-scale cases. Parameters are set as shown in Table I.

TABLE I
PARAMETERS OF GA

Parameter	**Value**
Retain rate	0.4
Random select rate	0.2
Population size	20
Mutation rate	0.8
Iteration	50

B. Results and discussion

The results of TWGA and F-TWS are shown in Table II. There are two sets of experiments for each case, the starting point and ending point of each AGV varies in these two different sets. Every experiment runs 10 times, to obtain the average optimal solution.

It can be seen in Table II that the optimal solution found by TWGA is better than that found by F-TWS, although F-TWS can find a solution in a shorter time.

For large-scale cases, the optimal solution of TWGA outperforms F-TWS a lot. For small-scale cases, the average optimal solutions of the two algorithms are much the same, and the calculation time of F-TWS is shorter. However, take the case with 30 AGVs as an example, the changing process of the optimal solution is shown in Fig.8. It can be seen from Fig.8 that the global optimal value is improved quickly by TWGA in the early stages of the iteration. When the number of iterations exceeds 20, the global optimal solution tends to be stable. Such performance indicates that the calculation time of TWGA can be optimized by adjusting the iteration time to a smaller number.

As for Automated Container Terminal, when ships reach port, the most important thing is to finish the loading and unloading tasks as soon as possible. Therefore, for both small-scale cases and large-scale cases in the real scenario, TWGA is more recommended owing to its better performance in the ability to find the optimal solution.

TABLE II
RESULTS OF THE EXPERIMENTS

Number of AGVs	**F-TWS**		**TWGA**	
	Average optimal solution	**Time**	**Average optimal solution**	**Time**
20-1	43	0.12	**41**	19.68
20-2	45	0.13	**43**	17.58
30-1	53	0.71	**50**	32.25
30-2	44	0.39	**43**	31.53
40-1	65	0.79	**59**	49.58
40-2	70	0.53	**58**	74.23
50-1	73	0.69	**62**	66.95
50-2	69	0.35	**60**	88.67

60-1	75	0.60	**63**	109.01
60-2	71	0.49	**63**	109.33
70-1	76	0.61	**68**	140.16
70-2	78	0.72	**67**	137.95
80-1	102	0.81	**78**	158.87
80-2	95	0.83	**73**	171.61
90-1	96	1.24	**78**	195.51
90-2	96	1.04	**79**	187.04
100-1	105	1.66	**81**	417.35
100-2	99	2.73	**79**	456.00
110-1	103	1.11	**87**	237.12
110-2	102	1.46	**83**	231.71
120-1	112	3.62	**90**	677.10
120-2	117	4.66	**88**	578.40
130-1	126	1.89	**95**	276.70
130-2	129	2.00	**92**	313.85
140-1	121	5.05	**90**	333.79
140-2	131	7.54	**103**	374.41
150-1	128	3.55	**108**	365.40
150-2	127	2.87	**102**	348.72
160-1	128	5.89	**97**	421.95
160-2	142	7.79	**100**	433.56

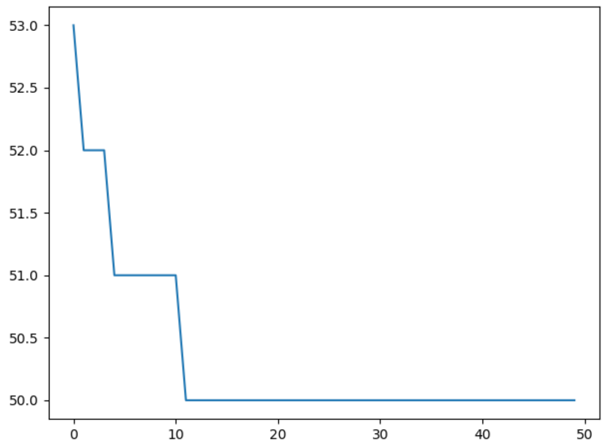

Fig. 8. An iterative process of the optimal solution for the case with 30 AGVs

V. CONCLUSION

In this work, the problem of conflict-free path planning is modeled as a min-max problem based on a directed graph. More specifically, the objective of the problem is to minimize the maximum completion time of all AGVs. To solve the problem, TWGA which combines conflict-free path planning and GA is adopted. FCFS policy and time window are used in the model and the algorithm respectively to achieve conflict avoidance. The results of the experiments show that the proposed method outperforms the given method F-TWS. In the future, the combination of the path planning problem and the AGV scheduling problem will be further study. The optimization of the calculation time will also be considered.

ACKNOWLEDGMENT

The author would like to thank the Artificial Intelligence Innovation and Development Project of the Shanghai Municipal Commission of Economy and Informatization under Grant 2019-RGZN-01057 for the funding support to this research.

REFERENCES

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

[2] M. Zhong, Y. Yang, Y. Dessouky, O. Postolache, Multi-AGV scheduling for conflict-free path planning in automated container terminals, Computers & Industrial Engineering 142 (2020) 106371.

[3] C. Wang, L. Wang, J. Qin, Z. Wu, L. Duan, Z. Li, M. Cao, X. Ou, X. Su, W. Li, Z. Lu, M. Li, Y. Wang, J. Long, M. Huang, Y. Li, Q. Wang, Path planning of automated guided vehicles based on improved A-Star algorithm, 2015.

[4] G. Qing, Z. Zheng, X. Yue, Path-planning of automated guided vehicle based on improved Dijkstra algorithm, 2017 29th Chinese Control And Decision Conference (CCDC), 2017, pp. 7138-7143.

[5] M. Zhong, Y. Yang, S. Sun, Y. Zhou, O. Postolache, Y.-E. Ge, Priority-based speed control strategy for automated guided vehicle path planning in automated container terminals, Transactions of the Institute of Measurement and Control 42(16) (2020) 3079-3090.

[6] Z. Fan, C. Gu, X. Yin, C. Liu, H. Huang, Time Window Based Path Planning of Multi-AGVs in Logistics Center, 2017, pp. 161-166.

[7] A. Tuncer, M. Yildirim, Dynamic path planning of mobile robots with improved genetic algorithm, Computers & electrical engineering 38(6) (2012) 1564-1572.

[8] Z. H. Sun, D. Liang, Z. Zhuang, L. Chen, X. Ming. Multi-task processing oriented production layout based on evolutionary programming mechanism. Applied Soft Computing 98 (2021) 106896.

[9] A. Alshamsi, A. Diabat. A Genetic Algorithm for Reverse Logistics network design: A case study from the GCC. Journal of Cleaner Production 151 (2017) 652-669.

[10] D. I. Arkhipov, D. Wu, T. Wu, et al. A parallel genetic algorithm framework for transportation planning and logistics management. IEEE Access 8 (2020), 106506-106515.

[11] Z. -H. Sun and X. Ming, "Multicriteria Decision-Making Framework for Supplier Selection: A Customer Community-Driven Approach," in IEEE Transactions on Engineering Management, doi: 10.1109/TEM.2021.3089279.

[12] Y. Zhou, W. Wang, D. He, Z. Wang, A fewest-turn-and-shortest path algorithm based on breadth-first search, Geo-spatial information science 17(4) (2014) 201-207.