Multi-AGVs Path Planning in Intelligent Parking Lot

Xiaoyong Liao, Yang Wang and Jingdong Lin

ABSTRACT

The problem of path planning for multiple Automated Guided vehicles (AGVs) in intelligent parking lot. Proposing a path planning algorithm that combining K shortest path with time window. Firstly, using the visibility graph algorithm to model the parking lot. According to the established model, the K short paths of each AGV are obtained by combining the A* star and path deviation algorithm. Then, the time window algorithm is used to predict the time that each AGV occupies path. If some AGVs’s time windows have overlapped, comparing the waiting strategy with the alternative K paths strategy, and the shortest strategy is adapted to solve the conflict. Finally, an example is given to verify the proposed algorithm, and the results show that the proposed algorithm can plan a collision-free path for each AGV in the intelligent parking lot of multiple AGVs. And the searching speed is relatively fast.

KEYWORDS
Intelligent Parking Lot, K Shortest Path, Time Window, Path Deviation.

INTRODUCTION

The rapid development of the world economy has led to more and more vehicles on the urban roads. But the limited number of parking spaces will lead to the difficulty of parking, and the illegal parking will make the traffic more crowded. So how to park more vehicles in the limited space has become an urgent issue in transportation. The emergence of automated guided vehicle (AGV) in intelligent parking lot has improved this problem more effectively, because of its small area, many parking lots and high degree of automation. However, in order to make the AGV accomplish assignment fast, and prevent conflicts during work, path planning must be implemented for multiple AGVs.

There are many methods for multi-AGVs path planning, but these are applicable to container terminals, intelligent warehouse, factories, etc. Bobanac et al. propose a modified Banker’s algorithm [1] for path planning of multi-AGVs in the factory. Sun et al [2], build a color, resource-oriented time Petri net to avoid the conflict and deadlock of multiple AGVs in the logistics center. Saidi-Mehрабad et al. [3] propose a two-stage ant colony algorithm to solve multi-AGVs scheduling in the workshop.
Zaghdoud et al. [4] mix Dijkstra algorithm, genetic algorithm and heuristic method to settle multi-AGVs path planning of container terminal. Sun et al. [5] combine Dijkstra algorithm with dynamic time window algorithm to solve multi-AGVs path planning in parking lot. It indicates that there are few works for multi-AGVs path planning in parking lots. What’s more, the general path planning algorithms are not suitable for intelligent parking lots, because the area’s space is limited, and the general algorithms provide only one path for the AGV initially. So when there is a conflict among AGVs, the path of AGV should be re-planned. As a result, AGVs work in an inefficient situation.

In this paper, the $K$ shortest path algorithm can provide $K$ paths for the AGVs. When some AGVs are in conflict by using the time window to judge, the optimal path can be quickly selected from the alternative $K$ paths.

PATH PLANNING ALGORITHM OF AN AGV

The Model of the Parking Lot

In order to make path planning for AGV, it is primary to build the parking lot model that represents the roads, parking lots and other information. The parking map is showed in Figure 1 Generally, there are three methods to model, the visibility graph, the grid method and the topological method. And the last one is used commonly [6], because it is easy to implement and has a visual effect. However, it can’t reflect the relationship between two nodes in parking lots, and it saves less information about the environment, but the visibility graph algorithm can conquer these defects. So this paper uses the visibility graph algorithm to model. The model of the intelligent parking is mainly composed of the parking station and path information, and these information is modeled by the Class object of C++, so the Class objects of the parking station and the Class objects of the path are constructed.

![Figure 1. The working environment of AGV.](image-url)
1) The Class object of the parking station: Figure 2 shows the parking station model, the Class object of each parking space has four pointers representing adjacent parking space in four directions of up, down, left and right. If an adjacent parking space doesn’t exist, the pointer of that direction is empty. Therefore, starting from any parking space, obtaining any parking space with pointers in four directions. What’s more, it also stores a large amount of information about the parking space including coordinates, the status, etc.. The intersections and the terminal of the road also use the same definition of this Class object, because the functions of these nodes are unified.

![Figure 2. The model of the parking station.](image)

2) The Class object of the path: Giving any two nodes and the length can determine a path. So the Class object of the path is composed of the node of the start, the end and length. What’s more, the node of the start and the end need the above definition of the Class object of parking space to achieve. The model of the path is showed in Figure 3 Each path must correspond to the start, the end and the length, and these can’t be empty. The constructions of the above models ignore the detailed data that are independent of the algorithm and the functions of the Class object.

![Figure 3. The model of the path.](image)

3) The combination of parking station and path: Figure 4 shows the model of the combination, Up1 represents the pointer of the upper parking, Up2 represents the pointer of the upper path, and the others represent almost the same meaning in addition to the direction. The path’s Class object is used to define the path that connects to the Class object of parking space in the up, down, left and right directions, which realize the combination between the parking station and the path.
The K Shortest Path

The problem of the K short paths [7] that finding the best path, the second best path ... the K-th best path from the start to target. In this paper, the A star algorithm and Yen algorithm are combined to realize the K shortest path. Firstly, using the A star heuristic search algorithm to obtain the shortest path \(P_1\) from the starting station to the target. Then solving the remaining \(K-1\) paths based on the \(P_1\), when calculating the \(i\)-th path \(P_i\), all nodes except the target on the path \(P_{i-1}\) are regarded as deviating nodes, and using A star algorithm to solve the shortest path from each deviating node to the target, and add it to the path from the start to the deviation node on the previous path \(P_{i-1}\) to form a candidate path. Comparing all the candidate paths, and select the path with the least cost that is \(P_i\). Figure 5 is the flow chart of the K shortest path.

![Diagram](image-url)
PATH PLANNING OF MULTI-AGVS

There is a little chance that each AGV reach the target according to the best path in its \( K \) shortest path, because of the limited space and the high working frequency of the multi-AGVs. This paper adapts the \( K \) shortest and time window algorithm to solve the conflict of multi-AGVs.

The Model of Time Window

In this paper, the time window refers to the total time spent by AGV that enter a path until leave. The time window of AGV shows in Figure 6, and the white area symbolize the free period of the current path, so the other AGV can pass it safely, while the black area are occupied period, and other AGVs can’t use that path during this time.

\[
\begin{align*}
\langle C_9, P_{126} \rangle & \quad \text{--} \\
\langle C_8, C_9 \rangle & \quad \text{--} \\
\langle C_{7,8} \rangle & \quad \text{--} \\
\text{The} & \quad \text{path} \\
\langle C_5, C_6 \rangle & \quad \text{--} \\
\langle C_3, C_4 \rangle & \quad \text{--} \\
\langle C_{1,2} \rangle & \quad \text{--}
\end{align*}
\]

The running time of AGV /s

Figure 6. The time window of AGV.

The Types of Conflict

The different types of conflict should apply different methods of path planning in order to improve the efficiency. There are three types of conflict during multi-AGVs driving.

1) The conflict of node

\[
\begin{align*}
\text{Figure 7. The conflict of node.}
\end{align*}
\]

Figure 7 shows the conflict of node, and that conflict means two or more AGVs arrive at a node at the same time.
2) The conflict of contradictory

![Figure 8. The conflict of contradictory.](image)

The conflict of contradictory is showed in Figure 8, this conflict means that two AGVs are driving in the same path with opposite direction at the same time.

3) The conflict in the same direction

![Figure 9. The conflict in the same direction.](image)

Figure 9 shows the conflict in the same direction, and the former AGV needs time to turn, so the latter AGV must collide, if it doesn’t turn and their distances are too short.

**The Strategies of Solving Conflict**

1) The situation of no conflict: If there are no conflicts in each path, each AGV should drive according to the best path of the K alternative paths.

2) The situation of conflict: There are two main methods to solve the conflict. Firstly, the lower priority of AGV stops running and waits until the higher priority of AGV leaves, and the sequence of tasks assigned to the AGV decides the priority, the early of the task, the higher priority it has. Secondly, the lower priority of AGV changes path and finds the conflict-free path from the K shortest path. If it doesn’t work, making a path change to the higher priority of AGV. The two methods must use the time window algorithm to detect until there is no conflict in each AGV.

Using the first method or the second method can figure out the conflict of node and the conflict in the same direction. Assuming that the first method spends $t_1$ to solve the conflict, and the second method spends $t_2$. If $t_1$ is smaller than $t_2$, choosing the first method to solve the conflict, reversely, choosing the second method.

Only using the second method to solve the conflict of contradictory.
SIMULATIONS

In order to verify the feasibility of the algorithm that this paper proposes, a program is written in Visual Studio software to simulate the paths planning of multi-AGVs.

The Simulation of Single AGV

In Figure 1, selecting the S5 as the start, and the P46 as the target. The value of $K$ in the $K$ shortest path is three. Assuming that AGV maintains a constant speed—$v=1\text{m/s}$ during driving, but it consumes ten seconds in the turning process. The results of simulation are showed in Figure 10, and the details of each path are showed in Table I.

![Figure 10. The simulation of single AGV.](image)

| Num. | The node of each path                                                                 | Time  |
|------|--------------------------------------------------------------------------------------|-------|
| 1    | S5$\rightarrow$C48$\rightarrow$C49$\rightarrow$C55$\rightarrow$C56$\rightarrow$C62$\rightarrow$C63$\rightarrow$C69$\rightarrow$C70$\rightarrow$C71 $\rightarrow$C72$\rightarrow$C73$\rightarrow$C74$\rightarrow$C75$\rightarrow$C76$\rightarrow$C77$\rightarrow$C65$\rightarrow$C58$\rightarrow$P46 | 71s   |
| 2    | S5$\rightarrow$C48$\rightarrow$C42$\rightarrow$C41$\rightarrow$C35$\rightarrow$C34$\rightarrow$C28$\rightarrow$C27$\rightarrow$C21$\rightarrow$C20 $\rightarrow$C1$\rightarrow$C2$\rightarrow$C3$\rightarrow$C4$\rightarrow$C5$\rightarrow$C6$\rightarrow$C7$\rightarrow$C8$\rightarrow$C23$\rightarrow$C30$\rightarrow$C37 $\rightarrow$C44$\rightarrow$C51$\rightarrow$C58$\rightarrow$P46 | 91s   |
| 3    | S5$\rightarrow$C48$\rightarrow$C49$\rightarrow$C55$\rightarrow$C56$\rightarrow$C62$\rightarrow$C63$\rightarrow$C69$\rightarrow$C70$\rightarrow$C71 $\rightarrow$C72$\rightarrow$C73$\rightarrow$C74$\rightarrow$C64$\rightarrow$C57$\rightarrow$C50$\rightarrow$C43$\rightarrow$C36$\rightarrow$C29$\rightarrow$C22$\rightarrow$C5$\rightarrow$C6$\rightarrow$C7$\rightarrow$C8$\rightarrow$C23$\rightarrow$C30$\rightarrow$C37$\rightarrow$C44$\rightarrow$C51$\rightarrow$C58$\rightarrow$P46 | 139s  |
The Time Complexity of this Algorithm

The maximum time complexity of the A start algorithm is \( O(m+n\log n) \) in the process of finding the shortest path, \( n \) is the number of nodes, and \( m \) is the number of paths. What’s more, it is necessary to iterate \( K-1 \) times when using the A star algorithm to find the \( K \) short paths. In each iteration, the path deviation algorithm is used to visit all the nodes of the previous path, so maximum complexity is \( O(n) \). As long as deviating from a node, a candidate path can be obtained by using the A star algorithm, and selecting the best path and deleting it from all the candidate path. The maximum complexity of this whole process is \( O(m+n\log n)+O(m)+O(\log m) \). So the time complexity of the algorithm that this paper proposed is

\[
O(m+n\log n)+
(k-1)[O(m)+(O(m+n\log n))+O(m)+O(\log m)]
\]  

(1)

The Simulation of Multi-AGVs

Assuming that there are three AGVs in the intelligent parking lot, named AGV1, AGV2 and AGV3, and the priority is from high to low. In Figure 1, selecting the S5 as the start of AGV1, the P32 as the target; selecting the S4 as the start of AGV2, the P124 as the target; selecting the S3 as the start of AGV3, the P44 as the target. Three AGVs launch at the same time. In Figure 11, getting the best path of three AGVs with \( K \) shortest path algorithm. And the time window of each AGV is showed in Figure 12.

Figure 11. The best path of the AGVs.
It can be clearly seen in Figure 12 that the AGV1, AGV2, and AGV3 are colliding in the path <C1,C20>. The running time of AGV1 is 22 seconds to 34 seconds in this path, the running time of AGV2 is 18 seconds to thirty seconds, and the running time of AGV3 is 14 seconds to 26 seconds. More information about this conflict is showed in Table II.

| Conflicting vehicle | Conflicting path | Starting time of conflict | Ending time of conflict |
|---------------------|------------------|---------------------------|------------------------|
| <AGV1,AGV2>         | <C1,C20>         | 22s                       | 30s                    |
| <AGV2,AGV3>         | <C1,C20>         | 18s                       | 26s                    |
| <AGV1,AGV3>         | <C1,C20>         | 22s                       | 26s                    |

The strategy of making a path change or waiting strategy to the AGV that has a lower priority, and the results are showed in Table III. As for the AGV2, it can clearly see that using the waiting strategy have a better performance than the strategy of the alternative path. While using the alternative path strategy for the AGV3 is better.
TABLE III. THE RESULTS OF DIFFERENT STRATEGY.

| AGV | The idea time | Waiting strategy consuming | Alternative path consuming |
|-----|---------------|---------------------------|--------------------------|
| AGV1 | 76s           | \                          | \                        |
| AGV2 | 46s           | 63s                       | 102s                     |
| AGV3 | 85s           | 102s                      | 97s                      |

After the conflict is resolved, modified paths of each AGVs are showed in Figure 13, and only AGV3 changes its path. What’s more, after adjustment, three AGVs do not have conflicts in their passing paths, as showed in Figure 14.

Figure 13. The modified paths of each AGV.
CONCLUSION

This paper studies the path planning of multiple AGVs in intelligent parking lots. It can not only provide $K$ paths for AGVs, but also predict the time and conflict of path through time window algorithm. When the conflict occurs, the AGV that has lower priority compares the strategy of alternative path with waiting strategy, and selects the better to solve the conflict. Finally, the algorithm is verified by simulation in Visual Studio software. The next plan will study how to select the target automatically so that the number of conflicts of multi-AGVs is minimized, and improve the parking efficiency.

REFERENCES

1. V. Bobanac and S. Bogdan . “Routing and scheduling in Multi-AGV systems based on dynamic banker algorithm,” 2008 16th Mediterranean Conference on Control and Automation, Ajaccio, 2008, pp. 1168-1173.
2. S. Sun, C. Gu, Q. Wan, H. Huang and X. Jia, “CROTPN Based Collision-Free and Deadlock-Free Path Planning of AGVs in Logistic Center,” 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 2018, pp. 1685-1691.
3. M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian and V. Mahmoudian, “An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs,” Computers & Industrial Engineering, vol.86, pp.2-13, Aug. 2015.

4. R. Zaghdoud, K. Mesghouni, S. Collart Dutilleul, K. Zidi and K. Ghedira, “A Hybrid Method for Assigning Containers to AGVs in Container Terminal,” IFAC Symposium on Control in Transportation SystemsCTS, vol.49, pp.96-103, May. 2016.

5. X. Sun, Y. Zhao, S. Shen, K. Wang, X. Zheng and Y. Shi, “Scheduling Multiple AGVs with Dynamic Time-windows for Smart Indoor Parking Lot,” 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)), Nanjing, 2018, pp. 864-868.

6. F. Jia, X. Liu, J. Wu, Y. Shi, F. Xu and Z. S. Zhang, “Research on Multi-AGV Autonomous Obstacle Avoidance Strategy Based on Improved A* Algorithm,” 2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Stuttgart, 2018, pp. 1-6.

7. C. Sub-r-pa and G. Chakraborty, “Dynamic distribute route recommendation system for multiple destinations,” 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Hua Hin, 2015, pp. 1-5.

8. W. Li and X. Su, “AGV path planning based on improved A* algorithm,” Modern Manufacturing Engineering, vol. 29, pp. 33-36, Oct. 2015.

9. S.R. Balseiro, I. Loiseau and J. Ramonet, “An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows,” Computers and Operations Research vol. 38, pp. 954-966, Jun. 2011.