

## 基于双向同步 A\*算法的 AGV 路径规划仿真研究

秦坤<sup>1</sup>, 曹冲振<sup>1</sup>, 李赫<sup>1</sup>, 鹿国庆<sup>1</sup>, 王洪祥<sup>2</sup>

(1. 山东科技大学 交通学院, 山东 青岛 266590;

2. 山东科技大学 机械电子工程学院, 山东 青岛 266590)

**摘要:** **目的** 解决 A\*算法在搜寻过程中查询的节点非常多、耗时较多等问题。**方法** 通过对传统 A\*算法进行改进, 得到改进双向同步 A\*算法, 使算法从起点和终点同时搜索路径, 并且对启发函数进行改进。为验证改进算法效果, 以仓储 AGV 为例进行二维与三维仿真。**结果** 改进双向同步 A\*算法在不同的地图下, 耗时分别减少了 39.1%, 34.3%, 34.6%, 搜索节点最多减少了 140, 路径长度基本一致。**结论** 改进双向同步 A\*算法较传统 A\*算法能够更加高效地规划出最优路径, 兼顾路径最优和出行复杂度低, 提高了检索效率, 能够有效解决查询节点多、耗时多等问题。

**关键词:** 双向同步 A\*算法; 启发函数; 路径最优; 出行复杂度

中图分类号: TP24 文献标识码: A 文章编号: 1001-3563(2019)21-0159-08

DOI: 10.19554/j.cnki.1001-3563.2019.21.023

### Simulation Research on AGV Path Planning Based on Bidirectional Synchronous A\* Algorithm

QIN Shen<sup>1</sup>, CAO Chong-zhen<sup>1</sup>, LI He<sup>1</sup>, LU Guo-qing<sup>1</sup>, WANG Hong-xiang<sup>2</sup>

(1. College of Transportation, Shandong University of Science and Technology, Qingdao 266590, China;

2. College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

**ABSTRACT:** The paper aims to solve the problems of A\* algorithm, such as a large number of nodes and time-consuming in search. The traditional A\* algorithm and the bidirectional synchronization A\* algorithm were improved, so that the algorithm can search the path from the starting point and the end point simultaneously, and improve the heuristic function. In order to verify the effectiveness of the improved algorithm, two-dimensional and three-dimensional simulation of warehouse AGV was carried out. The improved bidirectional synchronization A\* algorithm reduced the time-consuming by 39.1%, 34.3%, 34.6% under different maps. The searching nodes can be reduced by 140 at most, and the path length was basically the same. The improved bidirectional synchronization A\* algorithm is more efficient than the traditional A\* algorithm in planning the optimal path, considering both the optimal path and the low travel complexity, improving the retrieval efficiency, and effectively solving the problems of more query nodes and more time-consuming.

**KEY WORDS:** bidirectional synchronization A\* algorithm; heuristic function; path optimization; travel complexity

路径规划是 AGV (自动导引运输车) 实现自主工作的重要环节, 是实现 AGV 自主定位与导航的关键技术之一。路径规划的主要任务是为 AGV 规划出

一条无碰撞的运行路线, 保证 AGV 顺利完成搬运分拣任务<sup>[1]</sup>。全局路径规划是已知 AGV 工作环境的所有信息, 根据环境地图按照一定的算法搜索出一条接

收稿日期: 2019-03-25

作者简介: 秦坤 (1993—), 女, 山东科技大学硕士生, 主攻物流装备与控制技术。

通信作者: 曹冲振 (1975—), 男, 博士, 山东科技大学副教授, 主要研究方向为物流技术装备、移动机器人技术。

近最优的无碰撞路径<sup>[2]</sup>。

搜索路径的算法有多种，A\*算法是一种静态路网中求解最短路径最有效的直接搜索方法，是典型的启发式搜索算法，算法采用启发式的搜索方法可以减少搜索范围，控制搜索规模，降低问题的复杂度；A\*算法在存在多个最小值时无法找到最优路径，且在搜寻过程中查询的节点非常多、耗时较多。文中基于传统 A\*算法，改进算法从起点与终点同时搜索路径，直到到达最优中间结点时完成路径搜索，得到最优路径。为验证算法的有效性，以仓储 AGV 为例，利用 Visual Studio 仿真验证分析，利用 Flexim 软件测试双向 A\*算法在三维环境中的运行结果，最终得出结论：改进双向 A\*算法能够规划出一条最优路径，并且能够有效缩短计算用时。

### 1 地图模型

全局路径规划是指在环境已知的情况下进行规划，对 AGV 工作的环境进行地图建模。常见的地图模型有栅格模型、拓扑模型等。栅格地图容易构建与表示，并且表示的位置具有唯一性，对于仓库内部的短路径规划比较方便<sup>[4]</sup>，所以选择构建栅格地图。栅格地图直接与现场工作环境相对应，栅格地图栅格表示其实际可能的运行路径，每个栅格表示实际 1 m 的距离，地图模型中的黑色区域为障碍物信息，见图 1。

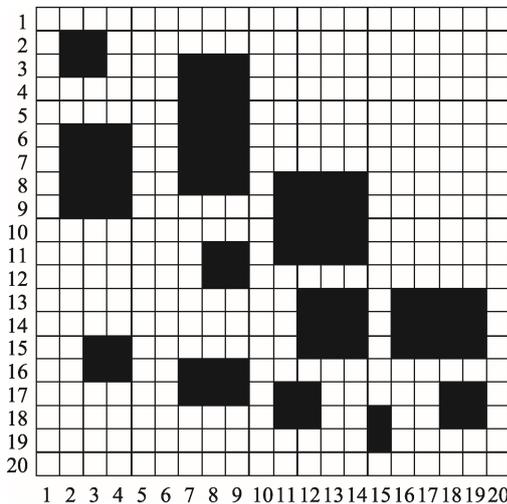


图1 栅格地图  
Fig.1 Raster map

### 2 标准 A\*算法

A\*算法是具有启发特征的搜索算法，算法比较灵活，能够适应多种环境，是静态环境中最为有效的路径规划算法<sup>[5-7]</sup>。A\*算法是通过将初始点与节点的代价以及节点到目标点的启发式评价，来分析目前的节

点。其代价估计函数为：

$$f(n) = g(n) + h(n) \tag{1}$$

式中： $f(n)$ 为从起始节点经由当前节点  $n$  到目标节点的估计代价； $g(n)$ 为初始节点到当前节点  $n$  的实际代价； $h(n)$ 为从当前节点到目标节点  $n$  的最佳路径的估计代价<sup>[7-8]</sup>。

常用的启发函数有曼哈顿距离函数、欧几里得距离函数等。如果地图环境中只允许朝上下左右 4 个方向移动，则选用曼哈顿距离函数，曼哈顿距离是指 2 个坐标点之间横轴和竖轴绝对值之和，当前节点为  $(x_n, y_n)$ ，目标节点为  $(x_s, y_s)$ ，则可以得到基于曼哈顿函数的启发函数：

$$h(n) = (|x_s - x_n| + |y_s - y_n|) \tag{2}$$

$$f(n) = g(n) + (|x_s - x_n| + |y_s - y_n|) \tag{3}$$

如果地图环境中允许朝 8 个方向移动，规划的路径允许斜线行驶，则选用欧几里得距离函数，欧几里得距离函数求得的为两节点之间的直线距离。启发式信息  $h(n)$ 为当前节点  $(x_n, y_n)$  和目标节点  $(x_s, y_s)$  之间的直线距离，则可以得到基于欧几里得函数的启发函数：

$$h(n) = \sqrt{(x_s - x_n)^2 + (y_s - y_n)^2} \tag{4}$$

$$f(n) = g(n) + \sqrt{(x_s - x_n)^2 + (y_s - y_n)^2} \tag{5}$$

在栅格地图中，AGV 不能全向移动，所以在实际应用的过程中需要进行转化，基于欧几里得的启发函数耗时较长，但是规划的路径更短。

### 3 双向同步 A\*算法

A\*算法在搜索路径的过程中，由于随着查询节点的增加，计算量随之增加，算法耗时增加，AGV 的响应时间较长；在搜索路径中更注重选择路径的最短距离，而忽略了 AGV 的响应速度。对传统 A\*算法进行改进，采用双向同步 A\*算法，从起点和终点同时搜索路径，能够有效减少算法耗时。对搜索过程进行创新，在搜索过程中不再把另外一个方向上产生的节点作为搜索的目标节点，而是将 2 个方向上当前节点的前一个节点作为 2 个方向搜索的目标节点，这样可以保证搜索节点同时更新，保证搜索的进行，当正向和逆向搜索同时将一个节点作为目标节点，且该节点满足 2 个方向搜索的约束条件，则搜索结束。

A\*算法的代价估计函数是影响算法效率的重要因素，代价估计函数的关键在于启发函数  $h(n)$  的选择， $h(n)$  的值越接近实际最优值，搜索的效率就越高。若  $h(n)=0$ ，则代价估计函数中只有  $g(n)$ ，此时为 Dijkstra 算法，遍历节点增加，降低了算法运行效率；若  $h(n)$  小于实际值但不为 0，此时  $h(n)$  越小，搜索的节点就越多，搜索效率就越低；若  $h(n)$  等于实际值，

此时算法运行速度最快，只搜索最优节点，此时选择的是最优路径；若  $h(n)$  大于实际值，此时算法不一定找得到最优值，所以启发函数的选择尤为重要。

由于 AGV 在转向过程中会对速度产生影响，并且 A\*算法在存在多个最小值时无法找到最优路径，综合考虑距离和转向角构造启发信息来改进启发函数，兼顾距离最短和出行复杂度低，对启发函数进行改进。引入最优路径指引函数  $f_1$ ，考虑当前考察点与目的地连线转向角，加大启发函数的权重，增加搜索的深度：

$$\theta = |\phi_1 - \phi_2| \quad (6)$$

$$f_1 = b^{\cos \theta}, b \text{ 为常数} \quad (7)$$

式中： $f_1$  为最优路径指引函数； $\phi_1$  为起点与终点的向量与  $x$  轴的夹角； $\phi_2$  为当前节点与终点的向量与  $x$  轴的夹角； $\theta$  为两向量的夹角。改进代价估计函数为：

$$f(n) = g(n) + f_1 * h(n) \quad (8)$$

基于双向同步 A\*算法 AGV 路径规划过程如下所述。

- 1) 初始化地图。
- 2) 输入搜索的起点信息与目标点信息。
- 3) 初始化每个节点信息，包括初始化  $f, g, h$  的值等为初始状态。把初始节点添加到正向搜索的 open 表中，把终点添加到逆向搜索的 open 表中。
- 4) 进行正向搜索。

① 在正向搜索的 open 表中取出最优估计距离节点，如果 open 表已空，则说明正向搜索了所有能搜索的节点还没有到达目标节点，则返回搜索失败结果，否则进行步骤②。

② 更新正向搜索的最佳路径节点。判断最优节点是否在逆向搜索的最佳路径中，若存在，则找到路径，若不存在，将节点添加到正向搜索 close 表中。

③ 对当前最优节点进行处理。找出当前节点的连接点  $m$ ，若  $m$  节点在正向 close 表中，则搜索下一个  $m$  点；否则判断  $m$  是否在正向搜索 open 表中，若不在分别计算  $g(m), h(m), f(m)$ 。

④ 设置  $m$  的父节点为  $n$ ，并把  $m$  按  $f$  值从小到大添加到正向搜索 open 表中。

⑤ 判断  $g(n)$  加上从  $n$  到  $m$  的距离是否大于  $g(m)$ 。若大于  $g(m)$ ，则搜索  $n$  的下一连接点，否则重新计算  $g(m), h(m), f(m)$ 。直到与  $n$  有连接的节点搜索完毕。

5) 进行逆向搜索，逆向搜索的步骤与正向搜索一致。

6) 判断步骤 4) 和步骤 5) 的返回值，如果返回 open 列表为空，仍未到达目标节点，则提示搜索失败状态，选择正向搜索的路径为最终路径；如果返回搜索完成标志，则显示完全路径。

## 4 仿真实验

对标准 A\*算法、双向同步 A\*算法在不同地图环境下进行对比仿真，仿真平台为 Visual Studio 2017，算法评价指标为：路径长度、路径规划结果、算法运行时间。首先通过 Visual Studio 实现算法，构造不同的仓库环境，并利用实现的算法进行路径规划。其次利用得到的数据，对算法的路径长度、运行时间进行对比分析，最后通过对比分析评价双向同步 A\*算法。

### 4.1 二维仿真实验

对仿真环境进行设置，设置障碍为对称型障碍，启发函数选用曼哈顿距离函数与欧几里得函数，在不同类型地图下利用标准 A\*算法对路径进行规划。地图 1 为多障碍物类型，地图 2 为单障碍物类型，红色为初始节点，绿色为目标节点，黑色为障碍物，蓝色为搜索过的节点但未加入到 close 列表，黄色为加入 close 列表但未被最终选用的节点，紫色为最终规划的路径。路径规划结果见图 2。

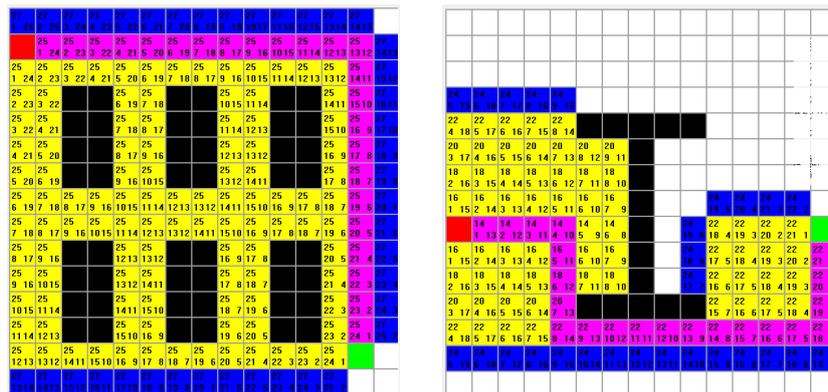
双向 A\*算法分别在地图 1 与地图 2 中进行路径规划，绿色为初始节点，红色为目标节点，黑色为障碍物，绿色为搜索过的节点但未加入到 close 列表，蓝色为加入 close 列表但未被最终选用的节点，紫色为最终规划的路径。启发函数选用曼哈顿函数，分别选用未改进启发函数与改进的启发函数，路径规划结果见图 3。

双向 A\*算法分别在地图 1 与地图 2 中进行路径规划，启发函数选用欧几里得函数，分别选用未改进启发函数与改进的启发函数，路径规划结果见图 4。

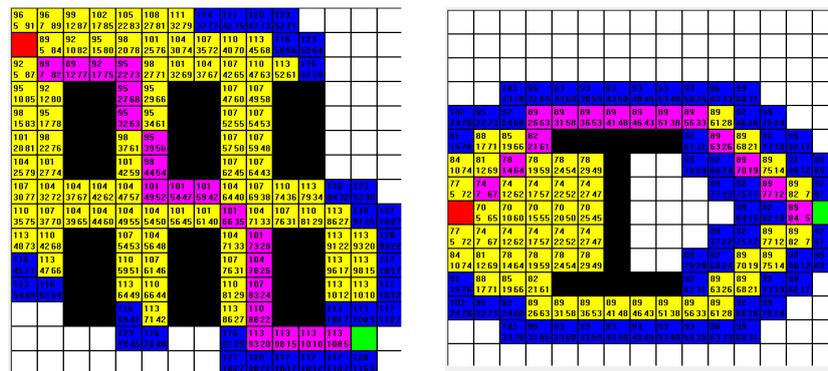
设置地图 3 为不对称障碍物环境，采用标准 A\*算法在地图 3 中对路径进行规划，分别采用曼哈顿距离函数与欧几里得距离函数，路径规划结果见图 5。采用改进启发函数双向同步 A\*算法在地图 3 对路径进行规划，结果见图 6。

对比标准 A\*算法与双向 A\*算法路径规划结果，规划出的路径基本一致，说明双向 A\*算法能够合理地规划出最优路径，并且减少算法的运行时间，提高算法效率。通过改进启发函数能够有效地降低路径的转折次数，并且搜索节点明显减少，能够有效地兼顾路径最短与出行复杂度低。

从仿真结果可以看出，使用欧几里得函数时，可以八方搜索节点，路径明显减少，并且搜索节点也相应减少。对 2 种算法的运行时间以及运行内存进行对比，以地图 1 为仿真环境，算法的启发函数采用欧几里得函数，对 A\*算法与双向 A\*算法进行运行时间以及运行内存测试，测试结果见图 7。



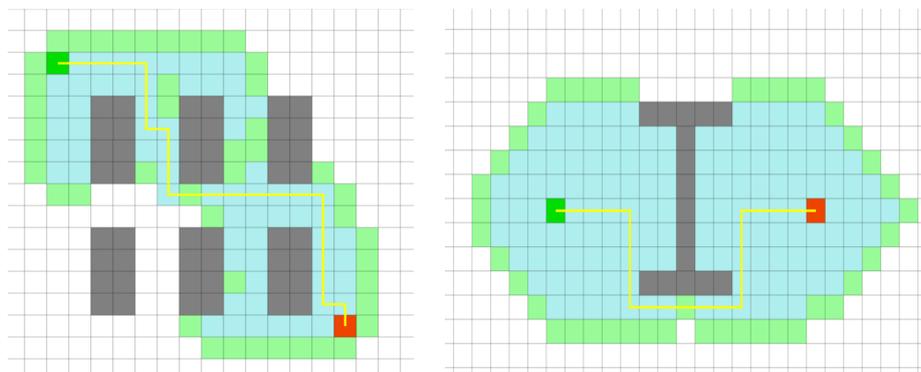
a 曼哈顿函数



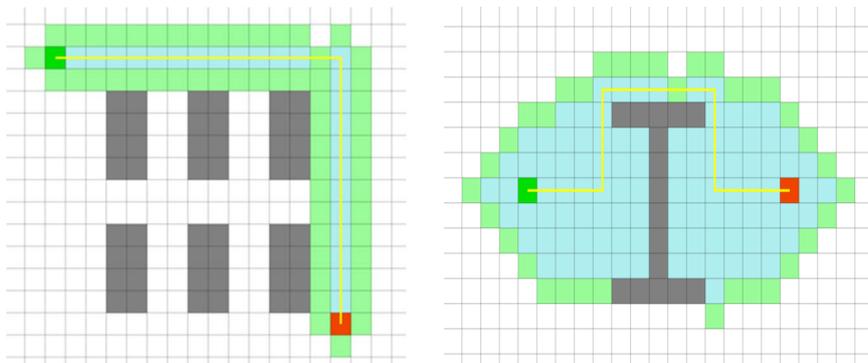
b 欧几里得函数

图2 标准A\*算法路径规划结果

Fig.2 Standard of standard A\* algorithm path planning



a 双向同步A\*算法



b 改进双向同步A\*算法

图3 基于曼哈顿函数的双向同步A\*算法路径规划结果

Fig.3 Result of bidirectional synchronization A\* algorithm path planning based on Manhattan function

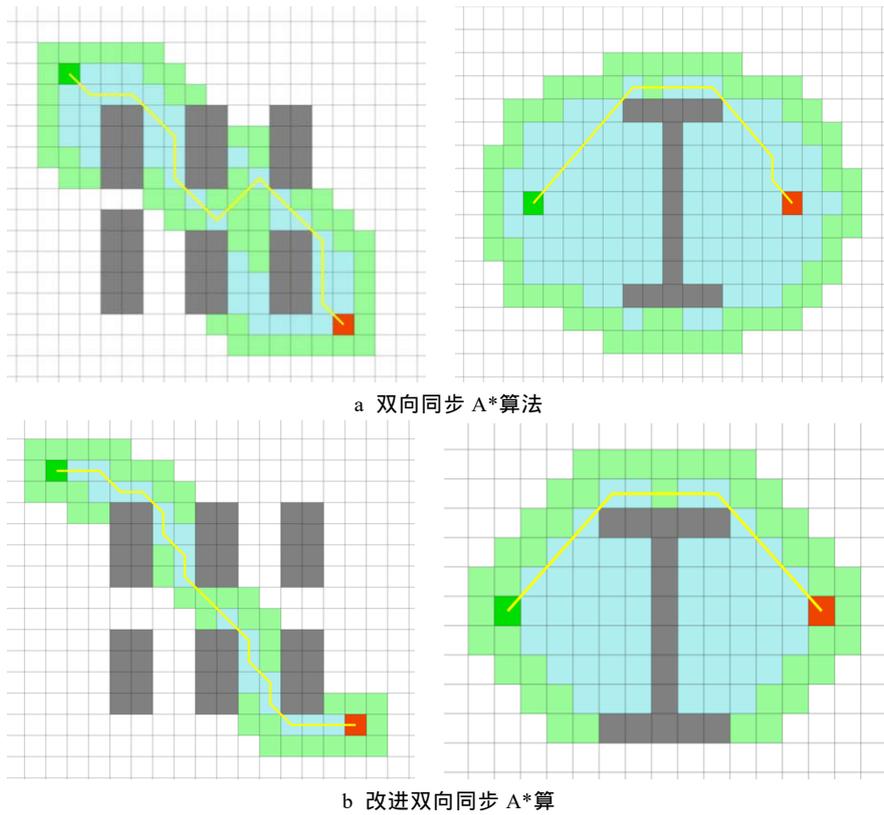


图 4 基于欧几里得函数的双向同步 A\*算法路径规划结果

Fig.4 Result of bidirectional synchronization A\* algorithm path planning based on Euclidean function

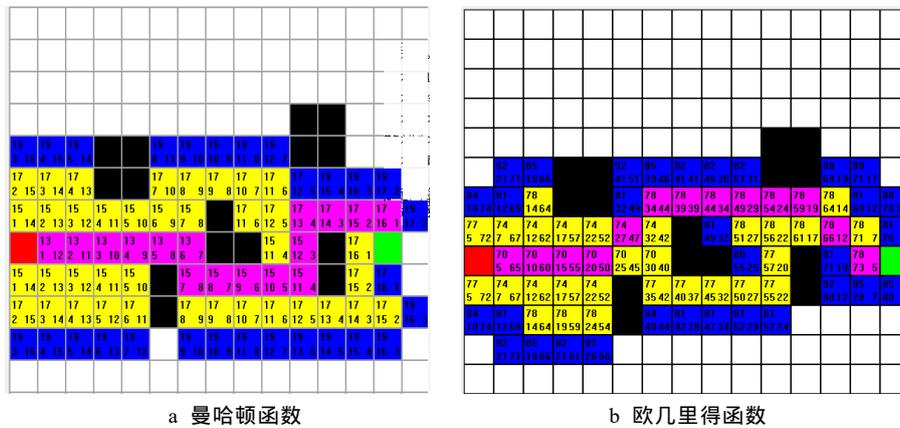


图 5 标准 A\*算法路径规划结果 (地图 3)

Fig.5 Result of standard A\* algorithm path planning (map 3)

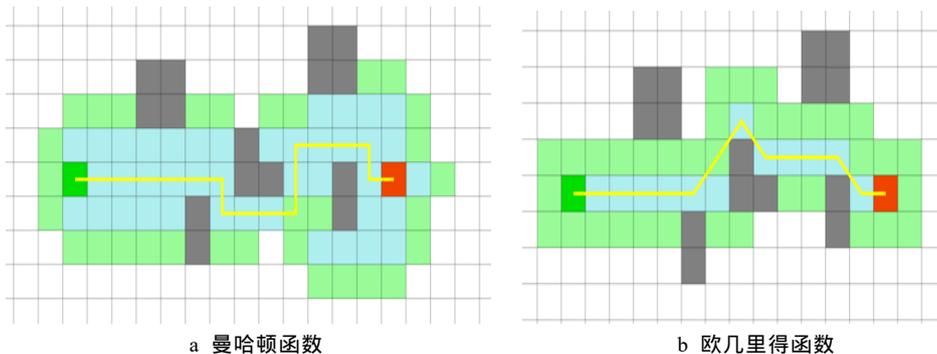
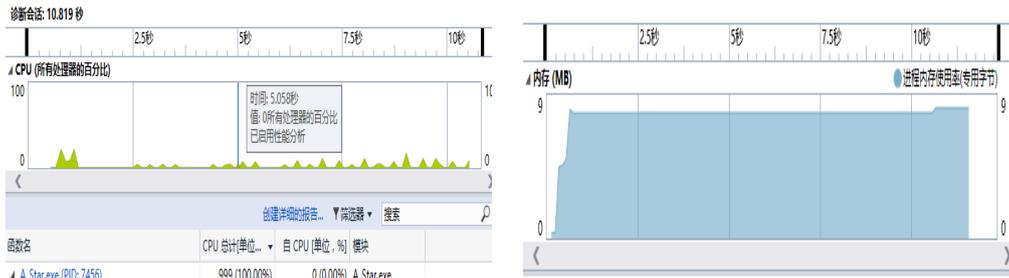
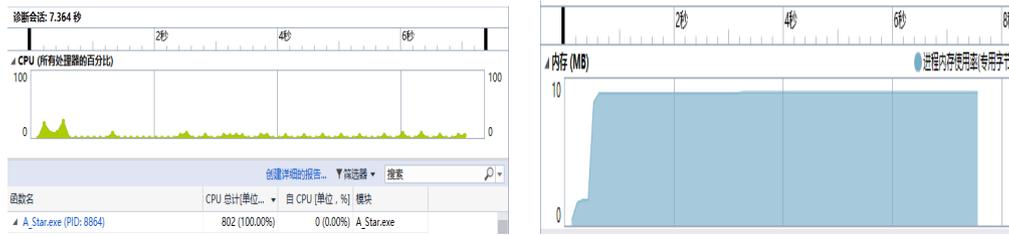


图 6 改进双向同步 A\*算法路径规划结果 (地图 3)

Fig.6 Result of improved bidirectional synchronization A\* algorithm path planning (map 3)



a 标准 A\*算法运行时间与内存



b 双向同步 A\*算法运行时间与内存

图 7 运行时间与内存

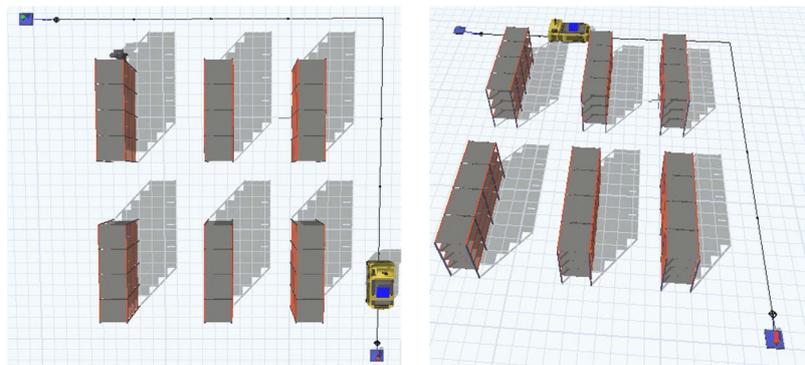
Fig.7 Running time and memory graph

从仿真结果来看，双向 A\*算法运行更为迅速，节省了 31.9%的运行时间，效果较为明显。双向 A\*算法占用内存量较标准 A\*算法大，但是增加量在合理范围内，因此双向 A\*算法优势非常明显。

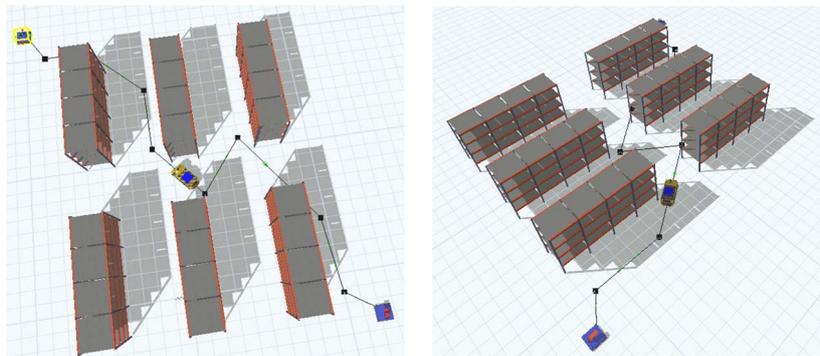
### 4.2 三维仿真验证

Flexsim 作为一个成熟的仿真手段，能够构建仓

库环境且仿真 AGV 运行情况，用来对算法进行验证<sup>[10-12]</sup>。以地图 1 为仿真环境，对双向 A\*算法进行三维仿真，启发函数分别选为曼哈顿距离函数与欧几里得距离函数。将障碍物设为货架，设置起点与终点，设定 AGV 空载速度为 2 m/s，负载速度为 1.5 m/s，最大加速度为 0.5 m/s<sup>2</sup>，三维仿真结果见图 8。



a 曼哈顿函数



b 欧几里得函数

图 8 三维仿真结果

Fig.8 Results of three-dimensional simulation

### 4.3 小结

根据仓库中实际货架与障碍物类型,选择对称型多障碍物环境、对称型单障碍物环境以及不对称障碍物环境,在这3种具有代表性地图环境下,以路径规划结果、算法运行时间及路径长度为评价指标进行仿真实验与结果分析,最终得出的结果见表1,包含2种算法的路径长度、运行时间与占用内存。

表1 仿真结果对比  
Tab.1 Comparison of simulation results

地图	标准 A*算法			双向 A*算法			结果对比 减少率/%
	路径 长度	运行 时间	搜索 节点	路径 长度	运行 时间	搜索 节点	
1	24	10.819	221	24	7.364	81	39.1
2	21	10.677	107	21	7.012	134	34.3
3	16	9.778	84	16	6.392	72	34.6

从表1可以看出,双向 A\*算法在运行效率上有极大的提高,能够有效减少搜索节点,极大地减少了运行时间,最高减少了39.1%的时间,最少减少34.3%的时间。经过在多种地图多次仿真,双向 A\*算法的运行时间减少了36%左右,搜索节点减少了54%左右。双向 A\*算法在保证路径最优的情况下,不仅有效并且能够减少搜索节点,还减少算法运算时间,改进的双向 A\*算法具有快速性、可靠性和稳定性。通过三维仿真结果,发现规划的路径折点处比较尖锐,不利于 AGV 的转向,算法需要进一步的改进,使得路径更加光滑。

## 5 结语

提出的改进双向 A\*算法,在保证搜索最优路径的情况下,能够平稳高效地运行,缩短 AGV 的响应时间,解决了 A\*算法在搜寻过程中查询的节点非常多、耗时较多等问题。故基于 A\*算法改进的双向时 A\*算法在路径规划结果、算法效率上,相较于2种算法都具有明显的优点,其路径规划结果可靠性高,可以合理地应用于 AGV 路径规划。下一步将算法运用到实际仓库场景中测试其性能,并且将进一步研究如何在保证最优路径的情况下使路径更加光滑。

### 参考文献:

[1] De GUZMAN M C, PRABHU N, TANCHOCO J M A. Complexity of the AGV Shortest Path and Single-loop Guide Path Layout Problems[J]. International Journal of Production Research, 1997, 35(8): 2083—2092.

[2] 赵江, 张岩, 马泽文, 等. 对 AGV 路径规划 A\*算法的改进与验证[J]. 计算机工程与应用, 2018, 54(21): 217—223.  
ZHAO Jiang, ZHANG Yan, MA Ze-wen, et al. Improvement and Verification of A-star Algorithm for AGV Path Planning[J]. Computer Engineering and Application, 2018, 54(21): 217—223.

[3] 杨超. 基于 A\*算法的机器人路径规划[J]. 电子技术与软件工程, 2015(24): 119.  
YANG Chao. Path Planning of Mobile Robots Based on A\* Algorithms[J]. Electronic Technology and Software Engineering, 2015(24): 119.

[4] 朱爱斌, 刘洋洋, 何大勇. 解决路径规划局部极小问题的势场栅格法[J]. 机械设计与研究, 2017, 33(5): 46—50.  
ZHU Ai-bin, LIU Yang-yang, HE Da-yong. Potential Field Grid Method for Solving Local Minimal Problems in Path Planning[J]. Mechanical Design and Research, 2017, 33(5): 46—50.

[5] 徐海军, 潘迪. 基于 A\*算法的无冲突路由多路 AGV 控制策略[J]. 工业控制计算机, 2018, 31(8): 100—101.  
XU Hai-jun, PAN Di. Conflict-free Routing Multi-channel AGV Control Strategy Based on A\* Algorithm[J]. Industrial Control Computer, 2018, 31(8): 100—101.

[6] ZHANG Y, LI L L, LIN H C, et al. Development of Path Planning Approach Based on Improved A-star Algorithm in AGV System[J]. International Conference on Internet of Things as a Service, 2017, 246: 276—279.

[7] 林娜, 李天啸. 基于双向 A\*算法的城市无人机航路规划[J]. 沈阳航空航天大学学报, 2016, 33(4): 55—60.  
LIN Na, LI Tian-xiao. Urban UAV Route Planning Based on Bidirectional A\* Algorithm[J]. Journal of Shenyang University of Aeronautics and Astronautics, 2016, 33(4): 55—60.

[8] BO Z, LINWEI L, YINGHAO Z. The Research on E-Commerce Logistics Picking AGV Path Optimization Method Based on the Improved A\* Algorithm[C]// International Conference on Cybernetics. IEEE, 2017.

[9] PENG J, HUANG Y, LUO G. Robot Path Planning Based on Improved A\* Algorithm[J]. Cybernetics & Information Technologies, 2015, 15(2): 171—180.

[10] 杨玮, 罗洋洋, 杨甜, 等. 基于 Flexsim 对某医药物流中心立体仓库优化[J]. 包装工程, 2017, 38(7): 210—215.  
YANG Wei, LUO Yang-yang, YANG Tian, et al. Based on Flexsim, a Three-dimensional Warehouse of a

- Medical Logistics Center Was Optimized[J]. Packaging Engineering, 2017, 38(7): 210—215.
- [11] JIA Y, JIANG P F. The Application of Simulation Technology in Distribution Center[J]. Applied Mechanics and Materials, 2017, 865: 675—680.
- [12] 陈元文, 吴晓波, 孙耀磊, 等. 基于 FlexSim 的军队自动化立体仓库货位优化研究[J]. 包装工程, 2013, 34(19): 74—77.  
CHEN Yuan-wen, WU Xiao-bo, SUN Yao-lei, et al. Research on the Optimization of Warehouse Location in Military Automated Three-dimensional Warehouse Based on FlexSim[J]. Packaging Engineering, 2013, 34(19): 74—77.
- [13] LIU T, FENG X. The Simulation and Optimization of Automated Storage and Retrieve System Based on Flexsim[C]// 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), 2012: 692—697.
- [14] 张原, 陈宇轩, 魏璐璐. 基于改进 A\*算法的 AGV 智能泊车算法[J]. 计算机系统应用, 2019, 28(1): 216—221.  
ZHANG Yuan, CHEN Yu-xuan, WEI Lu-lu. AGV Intelligent Parking Algorithm Based on Improved A\* Algorithm[J]. Computer System Application, 2019, 28(1): 216—221.
- [15] 陈豪, 李勇, 罗靖迪. 基于改进 A\*算法优化的移动机器人路径规划研究[J]. 自动化与仪器仪表, 2018(12): 1—4.  
CHEN Hao, LI Yong, LUO Jing-di. Path Planning for Mobile Robots Based on Improved A\* Algorithm Optimization[J]. Automation and Instrumentation, 2018 (12): 1—4.