

文章编号: 2095-2163(2023)11-0192-05

中图分类号: TP242

文献标志码: A

AGV 路径规划在智慧工厂中的应用

魏涛

(青岛理工大学 信息与控制工程学院, 山东 青岛 266525)

摘要: 随着现代化信息技术的发展,越来越多的生产制造企业进入了信息化、智能化的阶段。许多大型生产制造企业着手研究智能机器人,来代替传统的人工生产模式,进而越来越多地把注意力转向自动引导车(AGVs)。智慧工厂存储系统是现代制造业中的重要环节,AGV作为其中的关键设备,在运作过程中需要进行路径规划以保证效率和精度。因此,AGV路径规划是该领域热点研究方向。本文提出一种改进的A*算法进行无碰撞路径规划。利用栅格法描述自动引导车的运动轨迹,采用A*算法进行路径规划时,引入AGV共用路线的奖惩值,可以在降低路径冲突的同时优化预估开销。针对不同AGV冲突类型,制定不同路径碰撞解决规则,进一步提高路径规划的可靠性和效率。基于改进的A*算法和碰撞解决原则,快速规划AGV无碰撞路径,为智慧工厂的生产自动化提供坚实的技术支撑。使用C++语言开发AGV路径规划仿真软件,仿真软件实验结果表明:改进的A*算法可以提高AGV在智慧工厂中路径规划的运行效率,缓解多AGV系统运行过程中出现的交通拥堵现象。

关键词: 改进A*算法; 自动引导车; 路径规划; 智慧工厂; 栅格法

Application of AGV path planning in smart factories

WEI Tao

(School of Information and Control Engineering, Qingdao University of Technology, Qingdao Shandong 266525, China)

Abstract: With the development of modern information technology, more and more production and manufacturing enterprises have entered the stage of informatization and intelligence. Many large manufacturing companies are starting to research intelligent robots to replace traditional manual production modes, and are increasingly turning their attention to Automated Guided Vehicles (AGVs). The smart factory storage system is an important link in modern manufacturing industry, and AGV, as a key equipment, requires path planning during operation to ensure efficiency and accuracy. Therefore, AGV path planning is a hot research direction in this field. This article proposes an improved A* algorithm for collision free path planning. Using the grid method to describe the motion trajectory of the autonomous guided vehicle, and using the A* algorithm for path planning, introducing the reward and punishment values of AGV shared routes can reduce path conflicts while optimizing the estimation cost. Collision resolution rules are developed for different AGV conflict types to further improve the reliability and efficiency of path planning. Based on the improved A* algorithm and collision resolution principles, the collision free path of AGV is quickly planned, providing solid technical support for the production automation of smart factories. AGV path planning simulation software using C++ language is developed. The experimental results of the simulation software show that the improved A* algorithm can improve the operational efficiency of AGV path planning in smart factories and alleviate traffic congestion during the operation of multiple AGV systems.

Key words: improving A* algorithm; automatic guided vehicle; path planning; smart factory; grid method

0 引言

引入AGV后,传统的“人到货”的分拣方式转变为“货到人”的新模式,其优势在于可以通过自动化的方式,实现更高效、更智能的货物分拣生产模式。AGV的应用可以满足智慧工厂货物分拣的各种需求,带来了工作效率和精度的双重提升。因此,AGV在智能制造领域扮演了至关重要的角色,是实

现工业自动化不可或缺的关键技术之一。AGV对提高产业生产效率、减轻操作人员的负担、降低企业生产成本都具有非常重要的意义。路径规划是智慧工厂生产过程中实现货物运输的必要前提条件。在现代智能制造系统中,通过对AGV进行路径规划和任务调度,可以实现货物的智能化自动化运输,有效提高生产效率和产能,降低生产成本和风险。因此,深入研究AGV路径规划算法,优化和应用路径规划

作者简介: 魏涛(1993-),男,硕士研究生,主要研究方向:人工智能、大数据技术。Email:906168872@qq.com。

收稿日期: 2023-05-31

哈尔滨工业大学主办 ◆ 专题设计与应用

策略,对智慧工厂的生产运作具有至关重要的作用。关于 AGV 路径规划算法,目前有多种研究方法和实践应用,比如经典的 Dijkstra 算法、A* 算法、蚁群算法、遗传算法和模糊逻辑算法等。不同算法适用于不同的场景和问题,需要根据实际情况和需求进行选择和应用。房殿军等学者^[1]以电子商务行业仓储需求为基础,提出了自动化立体仓库中 AGV 的静态路径规划方法和动态避障决策策略。Mohammad 等学者^[2]研究一种高效的 2-opt 运算符,在 AGV 路径规划过程中用以解决路径重组问题,以此进一步提高路径规划的效率和准确性。在此基础上,还将 A* 算法与该运算符相结合,实现了多 AGV 线路的规划任务。Thomas 探究了一种局部搜索算符,并采用了蜂群优化的算法来解决 AGV 路径规划任务合理性分配问题^[3]。文献[4]利用局部搜索和随机搜索的策略来解决 AGV 系统调度和无碰撞路径规划问题。通过分析 AGV 系统调度和路径前瞻搜索算法结构,引入一种基于启发式的局部搜索算法来解决 AGV 调度问题。文献[5]基于蚁群算法提出 AGV 车间任务调度和无碰撞路径集成模型,有效地解决 AGV 任务分配和路径规划的问题,提高生产效率和安全性。

本文主要研究的是基于改进的 A* 算法在智慧工厂货物分拣系统中 AGV 如何进行路径规划。首先,利用栅格法描述 AGV 在智慧工厂货物运输过程。其次,为了解决多 AGV 在货物运输过程中带来的交通堵塞,通过在多 AGV 共同路线上加入惩罚值来改善交通拥堵情况。然后,结合改进的 A* 算法针对多 AGV 运行线路规划出一条无碰撞路径。最后,通过仿真实验对改进的 A* 算法的路径规划效率进行验证,并与其他算法的路径规划效率进行比较。

1 AGV 运行环境建模

研究中,可以把智慧工厂生产车间中 AGV 工作环境理想化视为矩形方阵。在环境建模的过程中,有多种常用的方法,其中包括拓扑图法、导航网络法以及栅格法等。以上 3 种方法都有其优缺点和适用范围。需要根据实际情况,选取最合适的方法进行建模,以保证建出的环境模型能够符合实际需求。本文采用栅格法来生成环境模型,栅格法环境建模如图 1 所示。

采用栅格法环境建模,其优点在于:

(1) 利用矩阵来近似模拟生产车间的工作环

境,可能规划出来 AGV 最短无碰撞路径并非最短路径,但可以增大 AGV 安全运行距离,提高安全性。

(2) 栅格法可对随机障碍域进行准确表达,不容易发生丢失障碍域等情况。

(3) 栅格法对障碍域的描述相对简单,从而可以提高 AGV 路径规划效率。

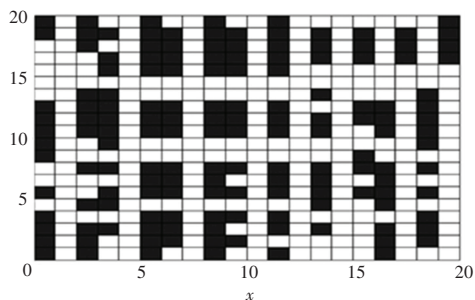


图 1 栅格法环境建模

Fig. 1 Grid based environmental modeling

2 改进 A* 算法的路径规划

2.1 A* 算法

A* 算法是一种启发式搜索算法,通常被用于寻找最短路径。是基于估价函数来指导搜索方向,在保证最优解的前提下,提高搜索效率和速度。其成本估计函数为:

$$f(n) = g(n) + h(n) \quad (1)$$

其中,函数 $f(n)$ 用于估计从起始节点通过任意节点 n 到目标节点的总成本。这个函数是由当前节点 n 的实际成本 $g(n)$ 和当前节点 n 到目标节点的最优路径估计成本 $h(n)$ 之和得出的。实际成本 $g(n)$ 表示起始节点到当前节点 n 的实际代价。当扩展节点 n 时,会将其所有邻居节点添加到 OPEN 表中,并将这些节点的 $g(n)$ 值设为 n 的 $g(n)$ 加上从 n 到对应节点的实际代价。这样就可以不断更新每个节点的实际成本,直至找到目标节点为止。 $h(n)$ 表示当前节点 n 到目标节点的最佳路径的估计成本。通常使用启发式函数来估算,例如曼哈顿距离、欧几里得距离等。启发式函数可以有助于在搜索过程中更好地指导搜索方向,从而更快地找到最优解。

A* 算法思想:A* 算法使用优先级队列来循环扩展最低成本估计节点,并把这个队列称为开集。在每一步执行过程中,从开集中删除 $f(n)$ 值最低的节点,然后更新它的相邻节点的 $f(n)$ 和 $g(n)$ 值,把这些节点添加到优先级队列中。A* 算法持续执行,直到找到目标节点,或者开集为空为止。最终的路

径长度是目标节点的值,该值总是比开集中所有节点的值更低。使用优先级队列可以确保在每一个步骤中按照 $f(n)$ 值递增的顺序,获取到开集中最低估计成本的节点,并且保证插入新的节点时能够自动维护开集中 $f(n)$ 值的最低节点。 A^* 算法实现步骤详见如下:

(1)定义节点类,包含节点坐标和距离起点的代价 $g(n)$ 以及到终点的启发式代价估计值 $h(n)$ 。

(2)初始化起始和终止节点,并将起始节点加入 OPEN 表中。

(3)在每个循环中,从 OPEN 表中选取 $f(n)$ 值最小的节点 n ,将其从 OPEN 表中移除,并将其加入 CLOSE 表中。

(4)检查节点 n 是否为终止节点。如果是,则停止搜索并返回从起点到终点的路径。

(5)如果节点 n 不是终止节点,则对其相邻节点进行以下操作:

① 对于每个相邻节点 m ,计算从起点到该节点的代价 $g(n)$ 加上从该节点到终点的启发式代价估计值 $h(m)$ 。计算得到相邻节点 m 的 $f(m)$ 值。

② 如果相邻节点 m 已经存在于 CLOSE 表中,则跳过该节点,并继续处理下一个节点。

③ 如果相邻节点 m 不在 OPEN 表中,则将其加入 OPEN 表中,并将节点 n 设置为相邻节点 m 的父节点,计算相邻节点 m 的 $g(m)$ 和 $h(m)$ 值。

④ 否则,如果新的 $g(m)$ 值比原来的小,则更新相邻节点 m 的 $g(m)$ 值和父节点,并重新计算 $f(m)$ 值。

(6)循环执行步骤(3)到步骤(5),直至找到终止节点或者 OPEN 表为空。

(7)如果 OPEN 表为空、并且未找到终止节点,则无法到达终止节点,搜索失败。

(8)如果找到了终止节点,并且从起点到终点的路径已经确定,则可以将该路径返回。

需要注意的是,在执行步骤(5)时,需要使用优先队列存储 OPEN 表中的节点,并总是选取 $f(n)$ 值最小的节点进行扩展。此外,启发式估价函数的准确性对 A^* 算法的性能和路径质量有着很大的影响。

2.2 A^* 算法的不足

(1)启发式函数不准确时, A^* 算法可能找到的不是最优解。 A^* 算法的性能高度依赖所使用的启发式函数的准确性,在某些情况下,因为启发式函数不准确, A^* 算法会找出次优解或者被卡在一些费时的路径上。

(2) A^* 算法可能会陷入局部最优解。在某些情况下, A^* 算法可能会受到其所使用的启发式函数的影响,沿着一个看似有趣、但是实际上并不符合题意的方向前进,最终陷入局部最优解。

(3) A^* 算法不适用于无向加权图。在无向图中,最短路径搜索的运行过程通常会出现节点被一次又一次地遍历的现象,这将明显增加该算法的复杂度,算法效率也将随之降低。

(4) A^* 算法对空间的要求较高。 A^* 算法需要维护一个优先队列来存储已经搜索过的节点。如果目标状态太大,可能会出现内存不足,从而导致算法无法运行。

(5) A^* 算法的实现较为复杂。在编写 A^* 算法时,除了需要合理选择启发式函数以外,还需要得到优先队列和开放列表等辅助工具。因此,需要花费一定时间来构建和调试算法。

综上所述, A^* 算法虽然在很多情况下表现优异,但也存在一些不足。因此,在具体应用时,需要综合考虑算法的优点和缺点,选择合适的算法并加以改进。

2.3 AGV 碰撞类型

在智慧工厂生产环境中,AGV 在运行过程中难免会出现碰撞情况,下面将介绍一些常见的 AGV 碰撞类型。

(1)节点冲突。节点冲突是指整个运行线路中,在某一时刻,有 2 个或多个 AGV 同时经过某一节点。节点冲突示意如图 2 所示。通过设置 AGV 路径优先级,可以避免发生节点冲突。

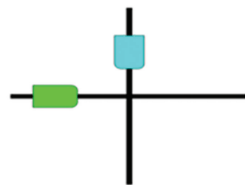


图 2 节点冲突

Fig. 2 Node conflict

(2)赶超冲突。赶超冲突是指 2 个 AGV 以不同的速度同向行驶。由于后车的速度大于前车的速度,两车的距离越来越近,如果保持这种状态,会导致两车发生追尾。赶超冲突示意如图 3 所示。



图 3 赶超冲突

Fig. 3 Catch up and surpass conflict

(3)相遇冲突。相遇冲突也叫做相向冲突。指 2 辆正在执行任务的 AGV 在同一条线路上相向而行。在同一时间节点,某条路线只能允许有一辆 AGV 通过,但如果此时有另外一台 AGV 也要通过该条路线,那么这 2 台 AGV 之间就会迎面发生碰撞。相遇冲突示意如图 4 所示。



图 4 相遇冲突

Fig. 4 Encounter conflict

2.4 A* 算法的改进

基于上述的 A* 算法的不足以及 AGV 路径碰撞类型,研究中提出对 A* 算法的改进。

在 A* 算法中,节点的 $g(n)$ 值是实际值, $h(n)$ 值是启发式估计值。选择适当的启发式估计函数能够提高算法的效率和寻找到更优路径。常用的启发式估价函数包括曼哈顿距离、欧几里得距离^[6-8]、对角线距离等。对此拟做阐释分述如下。

(1)曼哈顿距离。是根据标准坐标系中的绝对轴距总和来计算节点间的距离,相比于欧几里得距离,其计算简单、效率高,但是在寻找路径的质量上较低。

(2)对角线距离。在估价函数中考虑了沿对角线移动的情况,相对于曼哈顿距离和欧几里得距离,在进行评估值计算时与最优路径的评估值误差较小,因此可以有效提高路径的质量和算法的效率。

(3)欧几里得距离。是计算两点之间最短距离的一种方法,通常被用作启发式函数之一。但是,欧几里得距离计算涉及平方和开方运算,因此在寻找大型图形上的最短路径时,效率较低。

分析可知,选择合适的启发式函数取决于特定应用场景,通常需要平衡最短路径的质量和算法执行效率。在本文中,选择对角线距离作为启发式函数,因为该函数与最优路径的评估值误差较小,且计算相对简单,能够提高路径的质量和算法的效率。改进的启发式函数计算公式如下:

$$P = \min(|x_A - x_B|, |y_A - y_B|) \quad (2)$$

$$Q = |x_A - x_B| + |y_A - y_B| \quad (3)$$

$$h(n) = \sqrt{2c} \times P + c \times (Q - 2P) \quad (4)$$

其中,在栅格法地图中, c 表示水平或竖直移动一步的代价,而对角线移动的代价为 $\sqrt{2c}$ 。由式(2)~(4)可计算得到对角线距离的代价估值。

在同样的条件下,经计算对比发现,改进的 A* 算法运行效率得到了提高,同时扩展的节点并没有

减少,因此仍可得到与原 A* 算法相同的路径规划结果。

3 仿真实验

3.1 参数设置

在智慧工厂的模拟实验中,网格的大小为 $8 \text{ m} \times 8 \text{ m}$,工厂的长度和宽度均为 800 m 。AGV 数量为 100 台,AGV 的运行速度为 1 m/s 。为了设置适当的 x, y 参数,需要考虑每个参数的含义和对路径规划的影响。在该实验中,需要使距离越近的节点受到更多的惩罚,从而避免 AGV 出现拥堵或者交叉运行的情况。本文利用 C++ 语言开发智慧工厂中多 AGV 货物分拣的仿真软件。该软件可以监控 AGV 运动轨迹,调整 AGV 运行速度,通过栅格法建模,可根据系统需求自动设置工厂所要用的 AGV 数量。

3.2 AGV 防碰撞分析

由于本次仿真实验所有的 AGV 运行速度均为 1 m/s ,并规定后面的 AGV 停止等待前面的 AGV 通过后再继续运行。因此本次实验不考虑赶超冲突,只考虑节点冲突。多 AGV 防碰撞路径规划涉及到对多辆 AGV 路径的规划,可以采用改进的 A* 算法和蚁群算法。针对这 2 种算法,本文进行了实验比较,结果见表 1。

表 1 改进的 A* 算法与蚁群算法防碰撞对比

Tab. 1 Comparison of improved A* algorithm and ant colony algorithm in collision prevention

交叉口防碰撞 次数	模拟时间/ms		停止等待时间/s	
	蚁群算法	本文算法	蚁群算法	本文算法
1	30.63	44.20	6.48	5.81
2	124.88	89.31	8.99	8.32
3	314.57	135.03	14.13	13.46
4	794.38	181.56	22.43	20.78
5	1 187.17	226.91	33.84	31.09

从表 1 数据结果可以发现,随着 AGV 发生碰撞次数的增加,采用蚁群算法和改进的 A* 算法的路径规划模拟时间和停止等待时间也会随之增加。使用改进的 A* 算法进行多 AGV 防碰撞路径规划时,随着防碰撞次数的增加,模拟时间呈线性增长趋势。相比之下,蚁群算法的模拟时间则呈现出指数型增长,这是由于蚁群算法需要进行多次路径搜索和计算,而每次迭代都需要重新计算 AGV 的路径,尤其是在交叉口防碰撞时,其迭代次数就与碰撞次数相关,增加了计算复杂度。

本文中采用的改进 A* 算法在停止等待时间上稍优于蚁群算法。这是因为改进 A* 算法所用的网

格节点作为 AGV 的停止点,使得在交叉口防碰撞实验时,AGV 停止点与交叉口的距离更近,从而减少了等待时间,加快了 AGV 再次启动的时间。相比之下,蚁群算法的路径搜索过程比较复杂,在交叉口防碰撞时,需要较长时间进行路径搜索和计算,导致停止等待时间相对较长。因此,采用改进的 A* 算法进行路径规划可以有效缓解交通拥堵问题。

3.3 AGV 分拣效率分析

与传统 A* 算法相比,基于改进 A* 算法的多 AGV 路径规划能够更有效地提高 AGV 的路径规划效率,并进一步提升货物分拣系统的运行效率。A* 算法与改进 A* 算法分拣高效率对比见表 2。由表 2 可知,通过改进 A* 算法后,每秒平均分拣数(ASP/s)可以明显提高,这得益于在改进的 A* 算法中,优化了 AGV 路径的选取和优先级的调控,同时还加入了一系列的启发式函数和判定条件来提高搜索效率。这种多 AGV 防碰撞路径规划算法的优势在于对分拣系统整体性能的提升,可以更加高效、稳定地实现高速、大规模的货物分拣工作。

表 2 A* 算法与改进 A* 算法分拣效率对比

Tab. 2 Comparison of sorting efficiency between A* algorithm and improved A* algorithm

模拟次数/ 次	模拟时间/ s	传统 A* 算法 /(APS · s ⁻¹)	改进 A* 算法 /(APS · s ⁻¹)
1	1	3.54	3.85
2	2	3.43	3.79
3	3	3.32	3.78
4	4	3.22	3.75
5	5	3.12	3.75
6	6	3.01	3.74

从表 2 数据结果可以看出,在货物分拣系统中,无论是传统的 A* 算法、还是基于改进的 A* 算法,随着时间的增长,分拣速度都会逐渐减缓。然而,基于改进的 A* 算法在面对更多的货物时,可以更好地应对系统的需求,从而提高分拣速度,获得了更好的稳定性。这是因为基于改进的 A* 算法考虑了交通拥堵和防碰撞,能够规避道路拥堵和车辆碰撞的问题,提高了整个分拣系统的运作效率。因此,基于改进的 A* 算法进行多 AGV 路径规划可以提高分拣系统的整体效率和生产力。

4 结束语

本文针对智慧工厂中 AGV 路径规划采用栅格

法建模,基于传统的 A* 算法,提出了一种改进 A* 算法的路径规划,在不影响 AGV 运行效率的前提下,利用启发信息快速导向目标节点,能够实现 AGV 路径规划和主动避障,从而缓解交通拥堵和避免碰撞。未来工作应该针对不同的应用环境研发更加高效的路径规划算法。

参考文献

- [1] 房殿军,周涛. 自动化立体仓库中智能 AGV 群体的静态路径规划与动态避障决策研究[J]. 物流技术,2017(6):177-185.
- [2] MOHAMMAD M T, HALEH H, MOHAMMADI M. A mathematical model for scheduling of production process and allocation of an automatic guided vehicle in a flexible manufacturing system [J]. International Journal of Engineering Systems Modelling and Simulation,2018,10(2):125-131.
- [3] RYCK M D, VERTEYHE M, SHARIATMADAR K. Resource management indecentralized industrial automated guided vehicle systems[J].Journal of Manufacturing Systems,2020,54:204-214.
- [4] 姜利光,甘屹,孙福佳. 基于变参数阈值的随机扩展树算法路径规划研究[J]. 软件导刊,2017,16(12):67-69.
- [5] 李建军,许波桅,杨勇生,等. 多自动导引车路径规划的诱导蚁群粒子群算法[J]. 计算机集成制造系统,2017,23(12):2758-2767.
- [6] LIN Mingxiu, YUAN Kai, SHI Chenzhi, et al.Path planning of mobile robot based on improved A* algorithm[C]//Proceedings of Control and Decision Conference. Chongqing, China: IEEE, 2017:3570-3576.
- [7] 苏兰明. 基于案例推理的路径规划研究[D]. 保定:河北大学,2013.
- [8] 孟冲. 多 AGV 系统路径规划研究[D]. 杭州:杭州电子科技大学,2018.
- [9] 陈伟华,林颖,文宗明,等. 基于双重 A* 算法的机器人动态环境路径规划[J]. 组合机床与自动化加工技术,2018(4):127-130.
- [10] 赵晓,王铮,黄程侃,等. 基于改进 A* 算法的机器人路径规划[J]. 机器人,2018,40(6):903-910.
- [11] MARTIN J. A generic multi-layer architecture based on ROS-JADE integration for autonomous transport vehicles[J]. Sensors, 2018, 19(1):546-554.
- [12] HOWDEN W E. Cartography for navigation[J]. Proceedings of the IEEE, 1968,56(4):595-598.
- [13] CHEN W B, WU X B, LU Y. An improved path planning method based on artificial potential field for a mobile robot[J]. Cybernetics and Information Technologies, 2015, 15(2):181-191.
- [14] LISSOVOIA W W. Runtime analysis of ant colony optimization on dynamic shortest path problems [J]. Theoretical Computer Science, 2015,561:73-85.
- [15] RAHMAN H F, JANARDHANAN M N, NIELSEN P. An integrated approach for line balancing and AGV scheduling towards smart assembly systems[J]. Assembly Automation, 2020, 40(2):219-234.