文章编号:1671-4598(2023)10-0214-08 DOI:10.16526/j.cnki.11—4762/tp.2023.10.033 中图分类号:TP391.99 文献标识码:A

基于 WebGL 的多 AGV 物流调度仿真系统的研究与应用

荣 燕[1,2], 李 栋[3], 曹先庆[4]
(1. 沈阳化工大学 信息工程学院, 沈阳 110142;
2. 中国科学院 网络化控制系统重点实验室, 沈阳 110016;
3. 中国科学院 沈阳自动化所, 沈阳 110016;
4. 中国科学院 机器人与智能制造创新研究院, 沈阳 110169)

摘要: 针对 K 公司汽车总装生产线中多 AGV 电量分配不均、路径冲突、AGV 利用率低等物流调度问题, 为了提高 K 公司生产线中多 AGV 物流调度的效率, 设计了一种多 AGV 物流调度仿真系统; 利用 WebGL 技术建立了整个生产线场景的三维模型; 系统能够自动给 AGV 分配任务, 在 AGV 执行任务时, 将 AGV 的电池电量和 AGV 路径规划问题考虑到物流调度系统中; 该系统实现了对物流调度系统中 AGV 运输任务的合理分配, 提高了物流调度的效率; 经实际应用满足了 K 公司汽车总装生产线多 AGV 物流调度工程上的应用。

关键词: 电池电量; 多 AGV 分配任务; 路径规划; WebGL; 物流调度

Research and implementation of Multi-AGV Logistics Scheduling Simulation System Based on WebGL

RONG Yan[1,2], LI Dong[3], CAO Xianqing[4]
(1. School of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, China;
2. Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China;
3. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China;
4. Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China)

Abstract: Aiming at the problems of uneven power distribution, path conflict and low AGV (Automated Guided Vehicle) utilization rate in K company's Vehicle assembly line, a multi-AGV logistics scheduling simulation system was designed in order to improve the efficiency of multi-AGV logistics scheduling. WebGL (Web Graphics Library) technology is used to establish the three-dimensional model of the whole production line scene. The system can automatically assign tasks to AGVs, and take AGV battery power and AGV path planning into account in the logistics dispatching system when the AGVs perform tasks. The system realizes the reasonable allocation of AGV transportation tasks in the logistics dispatching system and improves the efficiency of logistics dispatching. The practical application satisfies the application of multi-AGV logistics scheduling engineering in the automobile assembly line of K Company.

Keywords: battery capacity; multiple AGVs assign tasks; route planning; WebGL; logistics scheduling

0 引言

目前生产型企业的数量众多, 每天都需要重复性搬运物料和产品, 耗时又耗力, 尤其是当比较忙碌时, 还容易出现送料错误的情况, 这些问题导致了工厂工作效率低下, 而且浪费了大量的人力成本。AGV (automated guided vehicle) 小车的出现, 可以轻松解决这个问题。AGV 不仅加速了生产进度和效率, 还可以节省大量的人力成本, 有效地减少了送错料混料的情况。针对多 AGV 任务调度的仿真研究, 很多研究考虑了多 AGV 执行任务时影响任务效率的因素。如 AGV 的电池电量、AGV 执行任务时的路径规划、AGV 数量配置等。文献 [1] 考虑了 AGV 的充电策略, 设计出 4 种充电方式, 得出合理的充电方式可以提高 AGV 的工作效率。文献 [2] 在集装箱码头作业过程中将电池电量的问题考虑在内, 并基于 Q 学习算法建立出多 AGV 动态调度模型: AGV 在集装箱码头作业时, 执行完一个任务时,

收稿日期:2023-03-23; **修回日期**:2023-04-23。
作者简介:荣 燕(1997-),女,硕士。
引用格式:荣 燕,李 栋,曹先庆. 基于 WebGL 的多 AGV 物流调度仿真系统的研究与应用[J]. 计算机测量与控制,2023,31(10):214-221.

可以动态选择下一个任务。文献［3］针对 AGV 路径优化，阐述了一种新的方法－改进人工势场法，解决了传统人工势场法中存在的局部极小值问题。文献［4］提出了用时间窗模型解决 AGV 路径问题，能够有效提高系统效率并且有良好的适应性。文献［5］以 AGV 完成任务的时间最短为目标，建立了同步和异步两种挑选模式下 AGV 的任务调度模型，最终验证了能够有效将解决 AGV 调度问题。文献［6］提出了路径优化速度较快的混合蚁群粒子群方法，提高了 AGV 运行效率。文献［7］研究了具有电池约束的 AGV 任务调度，以运输请求的延迟成本和 AGV 旅行成本的加权和最小为目标，将充电和运输请求分配给 AGV。文献［8］主要研究了一种新型的多品种小批量生产的矩阵式制造车间中的自动引导车辆调度问题。该问题的目的是确定一种解决方案，使客户满意度最大化，同时使分销成本最小化。为此，建设一个多目标混合整数线性规划的模型。此外也有很多研究路径冲突的文献，文献［9］针对传统时间窗法的局限性，提出了更好的多 AGV 路径冲突检测方法－基因测序算法。文献［10］针对 AGV 的冲突问题，提出模糊控制方法来优化 AGV 路径，证明了此方法可以有效提高运输效率。文献［11］提出的基于冲突搜索的多 AGV 路径算法最终验证能够有效解决路径冲突问题。文献［12］针对自动化码头动态作业环境中多自动引导车的路径规划问题，建立了 AGV 完成任务时间最小和路径无冲突的两阶段模型，可以有效解决多 AGV 动态冲突的规避。由上可见，大多数文献研究了 AGV 任务调度系统的算法、路径规划的算法和 AGV 充电策略，在任务调度系统中，很少有在研究的同时把 AGV 充电问题和执行任务时的路径规划问题考虑在内，因此将 AGV 电池电量和路径规划问题同时作为约束条件，研究适配 K 公司汽车总装生产线的实际场景需求是很有必要的，并利用 webgl 技术实现了三维 AVG 虚拟仿真[13-15]。

1　需求分析

AGV 是目前最受欢迎的自动化搬运设备，是实现智能物流的关键环节，从我国的 AGV 需求分析得知，AGV 的应用比较集中，主要分布在汽车工业和家电制造行业。AGV 越来越受欢迎的原因有很多，首先 AGV 小车不需要人工驾驶，只需要在 AGV 的中央控制系统的控制下即可自动接收和执行搬运任务，将需要搬运的东西自动搬运到指定的地点，真正意义上实现了整个生产过程中无人化、自动化地运送产品，大大提高了企业生产线的物流效率。其次，如今很多的企业都存在招工难的现象，人工成本在不断上升，因此很多企业都在向自动化模式转变，而 AGV 正好是可以代替普通工人的一种自动化搬运设备，降低成本，实现物流自动化，灵活性和准确性也都挺好，因而备受自动化制造业的欢迎。此外，智能制造时代的 3C 大批量定制，生产的周期也明显缩短，那么对于物流的速度也要求更高，因此企业对 AGV 小车的需求也会更加迫切[16-17]。

在 AGV 实际应用中，还存在着很多需要解决的问题：

1）为了提高多 AGV 调度系统的效率，很多研究提出改进算法来解决路径优化问题，却很少有学者研究小车充电对调度系统的影响，AGV 和电动汽车相似，充电所需时间都比较长，因此根据 AGV 电量安排任务就尤为重要。将 AGV 充电需求考虑到调度系统中是比较复杂的，AGV 每完成一次运输任务，都需要计算出小车此时剩余的电量，那么就需要知道小车行驶距离和所剩电量二者之间的关系，根据行驶距离计算出所剩电量，然后根据电量合理划分 AGV 充电区间，从而判断此 AGV 下一步是执行新的任务还是行驶到充点电充电[18-19]。

2）当越来越多的 AGV 被同时安排到仓库里工作时，避免 AGV 之间相互碰撞也成了研究的重点，目前大多数文献研究路径冲突检测的方法是时间窗法，但是时间窗法有一定的局限性，当 AGV 行驶的距离远，经过的路径节点多，那存储节点的时间窗就需要占据更大的内存，那么冲突检测的效率会降低，所以为了提高路径冲突检测的效率，本次研究利用基因测序算法来预测多 AGV 之间的路径冲突问题，但是由于基因比对和 AGV 路径问题的环境是不一样的，因此在环境建立方面有难度，不能完全复制到路径冲突检测当中[20-21]。

3）将充电需求和路径规划分别考虑到调度系统之后，最重要的是还需要将二者结合起来同时考虑到调度系统中，以最小路径为目标，建立多 AGV 模型也是一大难点。综上分析，本次研究以 AGV 充电需求算法、路径规划算法、调度系统模型为重点，分析了多 AGV 调度系统中问题的解决方法[22]。

2　AGV 充电需求算法分析

在物流运输行业，安全可靠的无人 AGV 广泛应用于工厂加工系统，具有低噪音、低污染、智能程控、高效搬运等诸多优点。驱动结构是 AGV 的重要组成部分，目前大多采用电池供电作为运行动力源。那么多 AGV 调度系统中电池的供电、放电和充电就显得尤为重要。本文主要研究调度系统中 AGV 的充电问题，以 K 公司汽车生产总装生产线为背景，共有十一辆 AGV 小车。小车负责把不同的汽车零件运送到终点进行组装，执行完任务后回到原点。当 AGV 一直处于工作状态，不停的在运输物料，AGV 的电量就会处于直线下降的状态，当 AGV 的电量不能够完成系统分配的运输物料任务时，此时 AGV 就要进入设定的充点电进行充电。等小车电量足够时，就停止充电，以保证运输任务的正常进行，从而减少任务等待的时间，提高 AGV 任务调度的效率。当 AGV 在完成运输任务时，存在两种状态，一种是空载，另一种是重载，AGV 同样的速度下，重载时的输出功率是要大于空载时的，相同的距离重载和空载的耗电量也有所不同，傅正堂等人提出了 AGV 电量消耗和距离的关系[23]。

根据图 1 得出，AGV 消耗的电量和路程的关系可以用

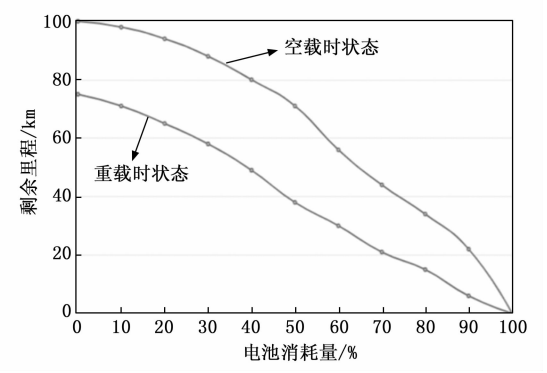

图 1 AGV 电量消耗图

一元二次函数描述，AGV 的电量记为 x，AGV 行驶的距离记为 R，二者关系记为：$R(x) = Ax2 + Bx + C$，因此根据 AGV 行驶的距离就可以知道小车所剩的电量。然后将 AGV 的电量分为 3 个区间：$x1$ 设为 AGV 从起点到终点最短距离所对应的电量，$x2$ 设为 AGV 能够从起点到达终点，再从终点到起点所行驶的总路程所对应的电量。

1）低电量区间：$[0 \sim x1)$，当 AGV 电量在这个区间时，说明 AGV 已经不能从起点到达终点了，此时 AGV 必须驶入充电点进行充电。

2）中电量区间：$[x1 \sim x2)$，当 AGV 电量处于此区间时，AGV 可以执行任务，但是在执行任务之前，需要考虑任务距离的远近，执行太远的任务中途可能会电量不足，从而导致小车停在半路，造成拥堵。所以在此区间的 AGV 尽量执行距离较近的任务。

3）高电量区间：$[x2 \sim 1)$，当 AGV 电量处于高电量区间时，可以执行任何一个任务。

通过划分 AGV 的电量区间，可以更加合理的给小车安排任务，提高系统调度的效率。小车执行任务的流程如图 2 所示。

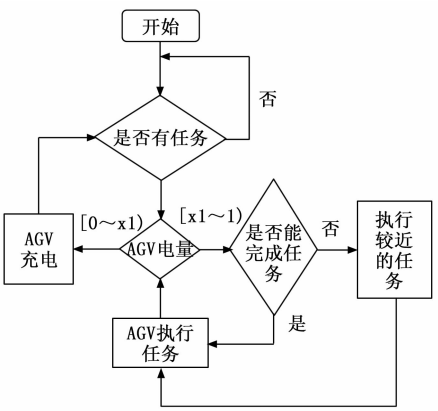

图 2 AGV 充电需求分析

3 AGV 路径规划

3.1 AGV 最短路径算法

在 AGV 路径规划的研究中，都以 AGV 路径最短为目标，完成这个目标一般使用的是 A* 算法。A* 算法的公式表示为：

$$f(n) = g(n) + h(n)$$

在研究 AGV 路径规划问题时，$f(n)$ 表示 AGV 从初始节点经过节点 n 再到目标节点的最小行驶距离，$g(n)$ 表示从初始节点到节点 n 的最小距离，$h(n)$ 表示从节点 n 到目标节点的路径的最小行驶距离，也称为 A* 算法的启发函数。如果启发函数 $h(n)$ 一直为零的话，此时就由 $g(n)$ 决定路径节点的优先级。如果 $h(n)$ 始终小于等于节点 n 到终点的距离，那么 A* 算法一定能够为 AGV 找到最短路径，但是当 $h(n)$ 的值越小，算法将遍历更多的路径节点，从而导致 A* 算法速度变慢。如果 $h(n)$ 完全等于路径节点 n 到终点的距离，则 A* 算法将找到最佳路径，并且速度很快。如果 $h(n)$ 的值比节点 n 到终点的距离要大，那么 A* 算法就不能保证可以为小车找到最短路径。由此可以得出，A* 算法找到最短路径的条件是：$h(n) \leqslant h(n)$ 具实距离。A* 算法（$F=G+H$）的具体步骤如下：

1）首先创建两个集合 openList 和 closeList（openList 用于存放可以走的路，closeList 存放已经走过的路（即不能走的路，包含障碍物））。

2）将起点加入 openList 中，使用一个 workCell 变量用于开拓道路。首先取出 openList 中的第一个元素存放到 workCell 中。此时进行判断，若 workCell 等于终点，流程结束。否则进行 3）。

3）将 openList 中的第一个元素即 workCell 从 openList 中删除，并加入到 closeList 中，表示这是走过的路。

4）找到 workCell 的所有邻点，用一个临时变量 neighboursList 存放，对 neighboursList [i] 进行判断，若该邻点在 closeList 集合中，表示该邻点是走过的，直接跳过，否则进行 5）。

5）若 neighboursList [i] 不在 openList 中，表示没有计算过 F，计算 F，并将 neighboursList [i] 的父节点设置为 workCell，然后将该邻居加入到 openList 中，加入的时候进行判断，当 F 最小时放在 openList 首位（保证最短路径）；否则，neighboursList [i] 在 openList 中（表示曾经计算过 F），执行 6）。

6）重新计算 F，若重新计算的值比之前计算的值小，则更新 F 和 G 以及父节点。

7）重复执行 2）～6），直到 openList 中没有元素（表示没有通路）。

8）若成功找到通路，则用终点来回溯寻找父节点，将每个父节点存到 pathList 集合中，最终该集合就是通路。

通过 A* 寻路算法可以算出 AGV 小车行驶的路径距离，下一步通过距离计算出小车到达节点的时间。AGV 在小车运动过程中，不会一直处于匀速状态，所以不能直接利用小车行驶的距离除以速度得出时间，需要考虑小车速度的变化从而求出时间。时间的计算方式主要如下：规定 AGV

的加速度为 a，最大速度为 V_{max}，匀加速运动和匀减速运动距离如式（1）和（2）所示：

$$S = V_0 t + \frac{1}{2}at^2 \tag{1}$$

$$S = V_0 t - \frac{1}{2}at^2 \tag{2}$$

当 AGV 小车从起点开始运动时，初始速度为 0，即 V_0 为 0，然后进行匀加速运动，加速到最大速度 V_{max}，此时能够求出从起点到达最大速度这段路程所需要的时间 t_1 为：

$$t_1 = \frac{V_{max}}{a} \tag{3}$$

将式（3）代入式（1），可得出从初始速度 0 加速至最大速度 V_{max}时所行驶的距离 S_{max}：

$$S_{max} = \frac{1}{2}at_1^{\,2} = \frac{1}{2}\frac{V_{max}^2}{a} \tag{4}$$

AGV 到达节点时间的求解主要分为 4 种情况：

1）AGV 到达节点时处于匀加速状态。AGV 从起点开始，初速度为 0，然后开始进行匀加速运动，到达节点时仍处于匀加速状态，从起点到达节点所行驶的距离为 S，根据式（1）可求出此时 AGV 到达节点的时间 t 为 $\sqrt{2s/a}$。

2）AGV 到达节点时处于匀速状态。如果 AGV 到达节点时处于匀速的话，说明 AGV 从起点开始行驶，先开始匀加速，然后加速到了最大速度 V_{max}之后，AGV 就保持 V_{max}进行匀速运动，所以这种情况下，AGV 到达节点的时间就是匀加速运动和匀速运动时间的总和，首先当 AGV 小车从起点开始运动时，初始速度为 0，即 V_0 为 0，然后进行匀加速运动，由于 AGV 到达下一节点时不需要转弯，行驶距离大于 $S_{max'}$所以能够加速到最大速度 V_{max}，此时根据式 3）就能求出 AGV 从起点到达最大速度时这一段路程的时间 t_1 为 V_{max}/a，下一步当 AGV 进行匀速运动时，一直以 V_{max}的速度行驶，这一段匀速行驶的距离记为 S_2，那么这一段路程所需要的时间为 $t_2 = S_2/V_{max}$。最后得出 AGV 到达节点处于匀速状态时的时间为 $t_1 + t_2$。

3）AGV 到达节点之前没有加速至最大速度，到达节点时处于匀减速状态。当 AGV 从起点开始运动，然后进行匀加速运动，由于 AGV 到达下一节点的距离较短，距离小于 S_{max}时就需要转弯，所以无法加速至最大速度就需要进行匀减速运动。当 AGV 从起点开始匀加速时，起始速度为 0，最终达到的速度 $V_g < V_{max}$，可得出这一段匀加速所需要的时间为 $t_1 = V_g/a$，当 AGV 进行匀减速运动时，初始速度为 V_g，那么根据式（2）可以求出这一路段所需的时间 t_2，最终到达节点的时间为 $t_1 + t_2$。

4）AGV 到达节点之前加速至最大速度，到达节点时处于匀减速状态。当 AGV 小车从起点开始运动时，初始速度为 0，即 V_0 为 0，然后进行匀加速运动，由于 AGV 到达下一节点时不需要转弯，行驶距离大于 $S_{max'}$所以能够加速到最大速度 V_{max}，此时根据式（3）就能求出 AGV 从起点到达最大速度时这一段路程的时间 t_1。到达最大速度以后，AGV 就进行匀速运动，一直以 V_{max}的速度行驶，这一段匀速行驶的距离记为 S_2，那么这一段路程所需要的时间为 $t_2 = S_2/V_{max}$，最后当 AGV 在下一节点需要转弯时，那么 AGV 的状态就会从最大速度开始减速进行匀减速运动，直到速度减为 0，这一段匀减速的路程记为 S_3，初始速度为 $V_{max'}$那么这一段的时间 $t_3 = V_{max}/a$ 。最终到达节点的时间 t 即为 $t_1 + t_2 + t_3$。

3.2　多 AGV 路径冲突检测算法

在 AGV 小车实际应用中，往往不是只有一台 AGV 在工作，而是同时有多台 AGV 在共同工作，并且物流运输系统中的各个任务都是交叉同时进行的。为了使多 AGV 系统能够高效有序的运行，需要为每个 AGV 规划出无冲突的路径，以免出现 AGV 之间发生碰撞或出现时间冲突。多 AGV 系统就是不但是一辆 AGV，而是很多辆同时工作，需要在保证 AGV 之间不发生碰撞的情况下完成系统分配好的任务。针对多 AGV 系统调度，我们首先要解决的问题是各个 AGV 之间碰撞问题和时间冲突问题，主要的冲突有 3 种类型，路口冲突、追赶冲突和相向冲突，分别如图 3～5 所示。

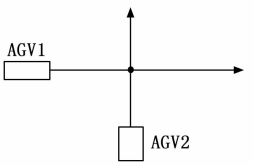

图 3　路口冲突

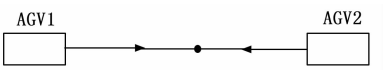

图 4　相向冲突

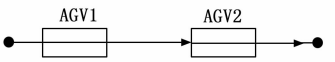

图 5　追赶冲突

由于时间窗法（就是使冲突路段和行驶时间窗口化，通常使用坐标轴进行表示。也就是说时间窗描述了每一辆 AGV 在地图模型中每个路段的驶入和驶出时间。）无法根据 AGV 冲突的类型调整预检测的精度，所以利用基因测序冲突检测方法[9]来解决路径冲突问题，能够有效弥补时间窗法的不足。算法的具体步骤如下：

1）对地图中的每一个节点和节点之间的路段进行编号。地图构成记为 $m \times n$，第一步对地图中的节点进行编号，表示为 0，1，2，…，$m \times n - 1$；水平路段编号：左节点为 i，右节点为 j，那么该路径的编号即为 $(i+j)/2$；垂直路径编号的方法和水平编号类似，路径上节点为 a，下节点为 b，则此路径的编号为 $(a+b)/2 + 0.01$。不同是垂直路径需要在后面加上 0.01，由于工厂仓库的规模一般都很大，水平垂直两个方向的路径编号很有可能会重复，因此垂直路径编号时多加 0.01 的数值，以便区分水平和垂直两个路段。

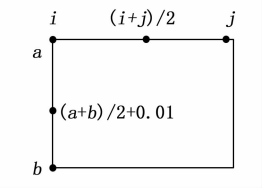

图 6 地图编号

AGV 的路径信息表示为 $R^i=\{P^i\ (T^i)\}$，其中 R^i 记为 AGV_i 路径，P^i 记为 AGV_i 经过的所有节点，T^i 为 AGV_i 经过节点的所有时间集合，经过节点的时间由 3.2 节的计算可得。其中，$P^i=\{\Phi_1^i, \Phi_2^i, \Phi_j^i, \cdots, \Phi_L^i\}$，$\Phi_j^i$ 表示 AGV_i 经过的第 j 个工位点（或路段）的编号；$T^i=\{t_1^i, t_2^i, t_j^i, \cdots, t_L^i\}$，$t_L^i$ 表示 AGV_i 经过工位点（或路段）Φ_j^i 的时间；L 表示 AGV_i 所行驶的距离。

2）建立 AGV 的路径和时间得分矩阵。

路径得分矩阵：路径得分矩阵记为 ***Mr***，矩阵 ***Mr*** 的行是一台 AGV 的路径，列是另一台 AGV 的路径，在基因序列比对中，相似性是指两个序列在同一位置具有相同的基因片段。和基因比对方法不同的是，AGV 行驶的路径是有方向的，所以就需要将两台 AGV 中的一台 AGV 的路径进行逆转。如果 AGV_1 经过的节点为 $[a1, \cdots, ai, aj, \cdots, an]$；$AGV_2$ 经过的节点为 $[b1, \cdots, bk, bl, \cdots, bm]$；那么可得出 AGV 之间的冲突发生在 $[ai, aj]$ 或者 $[bk, bl]$ 段。具体如图 7 所示：由于 AGV_1 和 AGV_2 的行驶的方向相反，所以 AGV_1 路线从左往右为 $a1 \sim an$，而 AGV_2 的路线从右往左为 $b1 \sim bm$。

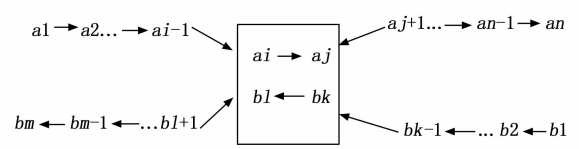

图 7 路径冲突段

假设 AGV_1 的路径信息为 R_1，路径节点为 P_1，路径长度为 L_1；AGV_2 的路径信息为 R_2，路径节点为则 ${P_2}^T$，路径长度为 L_2。***Mr*** 表示以（0，${P_2}^T$）为行，以（0，P_1）为列的（L_1+1）（L_2+1）的矩阵。

时间得分矩阵：时间得分矩阵记为 ***Mt***。由于时间没有方向性，所以不需要像路径那样做逆转处理。其余过程和 ***Mr*** 矩阵类似。假设 AGV_1 的路径信息为 R_1，时间序列为 T_1，长度为 L_1；AGV_2 的路径信息为 R_2，时间序列则为 T_2，长度为 L_2。***Mt*** 为以（0，T_2）为行，以（0，T_1）为列的矩阵。

3）回溯上面路径和时间得分矩阵，得到冲突发生的区域。

在基因里面，回溯是要找到相似性最高的片段，那么在 AGV 中，就是找到矩阵中的最大值。如图 8 所示，因此首先计算此元素上面、对角线方向、左面 3 个方向的得分，得分的数值为该方向上的得分＋移动过程得分；然后计算出的 3 个方向的得分与 0 的最大值。根据式（5）计算矩阵

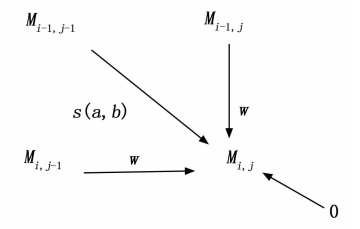

图 8 计算分

中每个元素的得分以此填充整个矩阵。

$$\boldsymbol{M}_{i,j}=max\{M_{i-1,j-1}+s(a,b), M_{i-1,j}+W, M_{i,j-1}+W, 0\} \tag{5}$$

其中：（$1 \leqslant i \leqslant L1$，$1 \leqslant j \leqslant L2$）

$M_{i,j}$ 为第 i 行第 j 列元素得分，$M_{i-1,j-1}$ 为对角线方向元素得分，$M_{i-1,j}$ 为上面元素得分，$M_{i,j-1}$ 为左边元素得分。s（a，b）和 W 为移动过程得分，L_1 和 L_2 表示 AGV_1 和 AGV_2 路径长度。

回溯的步骤：首先根据上面的计算方法得到矩阵中的最大值。下一步得到路径对比数组 $MAH\_r[]$ 和时间对比数组 $MAH\_t[]$，路径最大值数组 $MAX\_r[]$ 和时间最大值数组 $MAX\_t[]$。最后通过比对 4 个数组，得到路径比对结果和时间比对结果。

由于 AGV 小车一般不会出发点发生冲突，所以可以去除 $MAX\_r[]$ 和 $MAX\_t[]$ 第一个位置的元素，然后得到数组 $MAX\_r'[]$ 和 $MAX\_t'[]$，下一步计算出平均值，记 MAX_r_a 和 MAX_t_a。由于 AGV 行驶时会有延迟，所以设置了一个检测裕度 ε，然后在 $MAX\_r'[]$ 和 $MAX\_t'[]$ 中选择符合式（6）和（7）的元素并且存到 $mah\_r[\ ]$ 和 $mah\_t[\ ]$ 里面，具体如式（8）和（9）：

$$MAX\_r\_a-\varepsilon \leqslant MAX\_r'[] \leqslant MAX\_r\_a+\varepsilon \tag{6}$$

$$MAX\_t\_a-\varepsilon \leqslant MAX\_t[] \leqslant MAX\_t\_a+\varepsilon \tag{7}$$

$$mah\_r[i] \leftarrow MAX\_r'[i], 1 \leqslant i \leqslant min(L_1, L_2) \tag{8}$$

$$mah\_t[i] \leftarrow MAX\_t'[i], 1 \leqslant i \leqslant min(L_1, L_2) \tag{9}$$

然后取 $mah\_r[\ ]$ 和 $mah\_t[\ ]$ 的交集，就是两台 AGV 发生冲突的位置，如式（10）所示：

$$mah\_r[j] \cap mah\_t[k] \tag{10}$$

其中：j 和 k 的取值范围都为 $[1, min(L_1, L_2)]$。

多 AGV 冲突检测的伪代码如下所示：

```
冲突预检测：
定义：空列表存储有潜在路径冲突的 AGV 的数量：number[ ].
1：For i in range [1,n-1]
2：  For j in range [i+1,n]
3：If Si ? Sj≠ Φ
4：  number[ ] ←(i,j)
5：End For
6：End
```

冲突预检测之后确定路径冲突路段：如果不同的 AGV 行驶路径中包含相同的节点，比较 AGV 到达相同节点的时间，判断 AGV 是否在这路径中发生冲突。冲突路段的伪代码如下所示：冲突关键段：

定义：空列表存储存在路径冲突的 AGV 编号：section[ ]

空列表存储路径冲突关键段的长度：length[ ]

1：For i in range [1,k−1]

2：$[T_i]\leftarrow\frac{t_i^i+t_{i+1}^i}{2}$

3：[T] ←[T_i]

4：For j in range [1,2×(n−1)]

5：For k in range [j+1,2×n]

6：If [T_i] ⋂[T_k] ≠ Φ

7：section[ ] ←arg([T_i]或 [T_k])

8：length[ ] ←length[ ](section[ ])

9：End For

10：End For

11：End For

12：End

3.3　AGV 最终路径规划

上面两节介绍了 A* 算法求解最短路径，然后根据规划的路径进行冲突检测，检测完冲突以后再利用 A* 算法为冲突的 AGV 重新规划路径或者原地等待。具体实现过程如图 9 所示：图中有三辆 AGV，三辆 AGV 的路径具体经过哪些节点如表 3 所示。

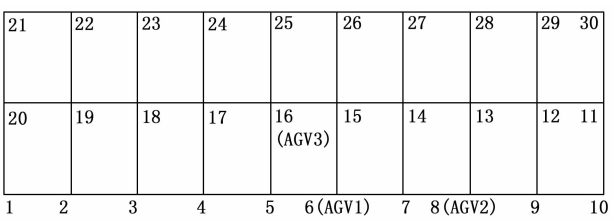

图 9　AGV 运行路径

表 3　AGV 路径节点

AGV	优先级	起点	终点	经过节点
AGV1	1	6	28	6→7→14→27→28
AGV2	2	8	26	8→7→14→27→26
AGV3	3	16	12	16→15→14→13

AGV 小车时间节点如表 4 所示。

表 4　AGV 时间节点

AGV1 时间节点			AGV2 时间节点		
节点	进入节点时间/s	离开节点时间/s	节点	进入节点时间/s	离开节点时间/s
6	0	1	8	0	1
7	2	3	7	2	3
14	4	5	14	4	5
27	6	7	27	6	7
28	8	9	26	8	9

由表 4 可知，AGV1 和 AGV2 在节点 7 处的冲突属于相向冲突，此时就为 AGV2 用 A* 算法重新规划一条路径，重新规划的路线为 8→9→10→11→30→29→28→27→26。由表 3 可知，AGV1 和 AGV3 在节点 14 处的冲突属于路口冲突，那么就让 AGV3 等待 AGV1 走完节点 14 再继续前行，从而避免冲突。

4　实验结果与分析

4.1　仿真原型系统搭建步骤

实现 AGV 三维模型利用 3 ds MAX 建立模型，3 ds MAX（3D Studio Max）是 3D 建模渲染和制作软件。首先建立三维工厂 AGV 模型。第二步是材质的设置，材质是指物体表面的特性，反应物体表面的质感。下一步是贴图的制作，贴图是材质的一种图像属性，为材质提供可视化的图像信息。然后是灯光的设置，灯光来源于观察和对画面的整体把握。全部设置好之后将建立好的模型以 gltf 格式导出。导出以后，再利用 WebGL（Web Graphics Library）将三维模型加载出来，WebGL 技术是可以在网页上绘制和渲染三维图形以及允许用户与之进行交互。渲染出来的效果如图 10 所示。

图 10　三维模型图

最终得到的多 AGV 模型如下：目标函数有两个条件，任务路径最短为 $r1$，当存在充电需求的 AGV 的时候，目标函数 r 为任务路径最短 $r1$ 和有充电需求的 AGV 路径最短 $r2$ 之和。目标函数如下：

$$r = r1 + r2 \tag{11}$$

$$r1 = min\sum_{i=1}^{I}\sum_{j=1}^{J}x_{ij}d_{ij} \tag{12}$$

$$r2 = min\sum_{m=1}^{M}\sum_{n=1}^{N}x_{min}d_{min} \tag{13}$$

约束条件如下：

任务调度车约束：

$$\sum_{i=1}^{I}\sum_{j=1}^{J}x_{ij} = I \tag{14}$$

一辆 AGV 一个时间只能完成一个任务：

$$\sum_{j=1}^{J}x_{ij} \leqslant 1 \tag{15}$$

如果第 m 辆 AGV 有充电需求，可以表现为当前电量处于低电量区间：

$$e_i \leqslant x1 \tag{16}$$

一辆 AGV 只能在一个充电桩内充电：

$$\sum_{n=1}^{N} X_{min} \leqslant 1 \tag{17}$$

式中，i 表示 AGV 需要完成的运输任务，$i = 1, 2, 3, \cdots, I$。j 表示 AGV 小车，$j = 1, 2, 3, \cdots, J$。表示任务 i 由第 j 辆小车执行，那么 $X_{ij} = 1$，否则为 0。d_{ij} 表示第 j 辆 AGV 小车在完成运输任务 i 过程中行驶的距离。

AGV 物流调度系统仿真实现如下所示：图 11 为调度系统的首页，记录了 AGV 的数量以及离线和在线的情况。图 12 为公司订单的一些信息统计，可以对订单信息进行增删改的操作。

图 11 系统首页

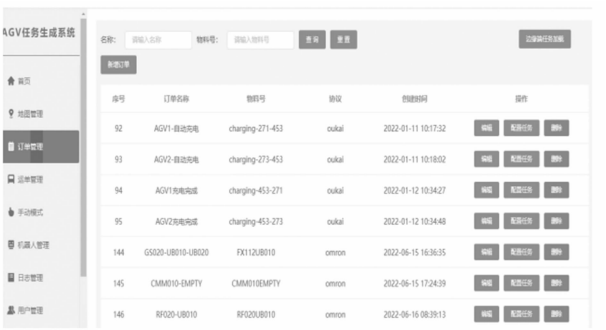

图 12 订单信息

4.2 结果分析

对于多 AGV 物流调度仿真系统的研究，都是为了提高系统中 AGV 的利用率、AGV 负载率等。但是不能单用上面一种指标来评价系统整体效率，不同的系统对于 AGV 的要求也是不一样的，所以评价多 AGV 调度系统的效率需要对系统进行全面的分析。

本文设计的系统以一天的工作时间进行仿真，在初始状态，所以 AGV 都在原点处，处于空闲状态。根据仿真时间，计算出 AGV 的利用率、负载率、平均电量等信息，原有调度系统和本文设计的系统数据对比如表 5 所示。

表 5 仿真结果对比 %

调度系统	AGV 利用率	AGV 负载率	AGV 平均电量
原有	60	50	40
现在	65	60	45

其中，一天中 AGV 的电量变化如图 13 所示。

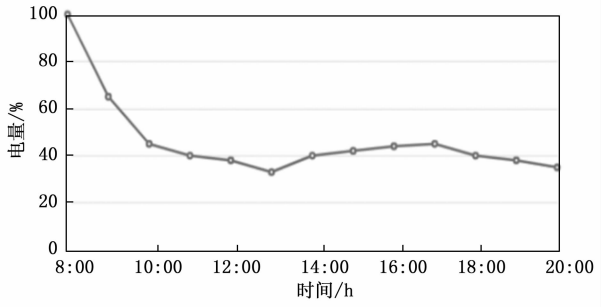

图 13 AGV 电量变化

通过表 5 可以得出，本文设计的调度系统由于考虑了 AGV 电量和 AGV 路径冲突，调度系统的任务分配更加合理，从而提高了 AGV 的利用率和负载率，提高了系统的总体效率。

5 结束语

本文主要以 K 公司汽车总装生产线为应用背景进行了研究，为了解决多 AGV 调度系统中电量分配不均、路径冲突等问题，利用 WebGL 技术建立了整个生产线场景的三维模型，并且建立了三维 AGV 模型，使得汽车生产线的场景和 AGV 小车更加可视化。此外，设计出了多 AGV 物流调度的仿真系统，系统可以自动给 AGV 分配任务，然后 AGV 按照系统里面派发的任务去完成。在 AGV 执行任务过程中，为了提高系统效率，设定了 AGV 电池电量的阈值，能够合理分配电量，并利用 A^* 算法为 AGV 寻找最短路径，然后再利用基因序列比对算法找到存在路径冲突的 AGV，并为它们重新规划路径。最终经过调度系统仿真验证，综合考虑电池电量和路径规划问题提高了多 AGV 物流调度系统的效率。

参考文献：

[1] 陈 珲，韩晓龙．考虑充电策略的自动化码头 AGV 调度 [J]．上海海事大学学报，2021，42 (2)：20－25.

[2] 邹梦艳．考虑电池电量约束的自动化码头 AGV 调度研究 [D]．武汉：武汉理工大学，2020.

[3] 牛秦玉，李美凡，赵 勇．改进人工势场法的 AGV 路径规划算法研究 [J]．机床与液压，2022，50 (17)：19－24.

[4] 泰应鹏，邢科新，林叶贵，等．多 AGV 路径规划方法研究 [J]．计算机科学，2017，44 (S2)：84－87.

[5] 袁瑞萍，王慧玲，孙利瑞，等．基于物流 AGV 的“货到人”订单拣选系统任务调度研究 [J]．运筹与管理，2018，27 (10)：133－138.

[6] 魏宏源，茅 健．基于混合蚁群粒子群的多 AGV 路径规划研究 [J]．制造业自动化，2019，41 (6)：59－61.

[7] SINGH N，DANG Q V，AKCAY A，et al. A matheuristic for AGV scheduling with battery constraints [J]．European Journal of Operational Research，2022，298 (3)：855－873.

[8] ZOUW Q，PAN Q K，WANG L，An effective multi-objective evolutionary algorithm for solving the AGV scheduling problem

with pickup and delivery [J]. Knowledge-Based Systems, 2021, 218: 106881.

[9] 张　政. 大型自动化仓储系统 AGV 在线无碰撞路径规划方法研究 [D]. 北京：北京化工大学，2022.

[10] 宋　宇. 一种针对自动化码头 AGV 拥堵冲突的动态路径优化策略 [J]. 港口装卸，2021 (4)：1－5.

[11] 郭　超，陈香玲，郭　鹏，等. 基于时空 A～＊算法的多 AGV 无冲突路径规划 [J]. 计算机系统应用，2022，31 (4)：360－368.

[12] 丁　一，袁　浩，方怀瑾，等. 考虑冲突规避的自动化集装箱码头 AGV 优化调度方法 [J]. 交通信息与安全，2022，40 (3)：96－107.

[13] CHAWLA V K, CHANDA A K, ANGRA S, et al. Simultaneous dispatching and scheduling of multi-load AGVs in FMS-A simulation study [J]. Materials Today: proceedings, 2018, 5 (11): 25358－25367.

[14] WITCZAK M, MAJDZIK P, STETTER R, el al. Multiple AGV fault-tolerant within an agile manufacturing warehouse [J]. IFAC-Papers OnLine, 2019, 52 (13): 1914－1919.

[15] LI N, ZHANG Y, HE F X, el al. Review of lithium-ion battery state of charge estimation [J]. Global Energy Interconnection, 2021, 4 (6): 619－630.

[16] LIAN Y D, XIE W, ZHANG L W, A probabilistic time-constrained based heuristic path planning algorithm in warehouse multi-AGV systems [J]. IFAC-Papers OnLine, 2020, 53 (2): 2538－2543.

[17] 于赫年，白　桦，李　超. 仓储式多 AGV 系统的路径规划研究及仿真 [J]. 计算机工程与应用，2020，56 (2)：233－241.

[18] 年田甜. 无人仓储环境下考虑充电需求的 AGV 路径优化研究 [D]. 北京：华北电力大学，2020.

[19] 陈香玲，郭　鹏，温　昆，等. 考虑充电需求和时间窗的多 AGV 调度优化建模 [J]. 河北科技大学学报，2021，42 (2)：91－100.

[20] 张祥祥，杨智飞，胡祥涛，等. 面向智能制造车间的 AGV 系统调度算法设计 [J]. 智能制造，2019 (9)：40－44.

[21] 陈　洋，林其岳，邓志华，等. AGV 路径规划算法研究进展 [J]. 机电技术，2022 (5)：39－43.

[22] 初鹏祥. 多 AGV 仓储系统任务调度及路径规划研究 [D]. 秦皇岛：燕山大学，2022.

[23] 傅正堂，胡志华，宗　康. 集装箱码头 AGV 电量非饱和状态下的调度优化 [J]. 大连海事大学学报，2017，43 (3)：58－62.

(上接第 213 页)

[2] LIU Z Z. Model and simulation validation based on the data of the aero experimentation [J]. Journal of System Simulation, 2002, 14 (3): 281－284.

[3] TIAN F Z, HUANGPU D H, SHA Y Z. Summary of the development of a new generation of airborne data acquisition system [J]. Measurement and Control Technology, 2007, 26 (3): 16－18.

[4] 马卿云，季航旭，赵宇海，等. 一种分布式异构带宽环境下的高效数据分区方法 [J]. 计算机研究与发展，2020，57 (12)：2683－2693.

[5] 丁　岩，杨万祥，汪　清，等. 大数据统一 SQL 引擎研究与设计 [J]. 科技视界，2019 (29)：1－4.

[6] ZHOU Z H. Using heuristics and genetic algorithms for large-scale database query optimization [J]. Journal of Information and Computing Science, 2007, 2 (4): 261－280.

[7] GARCíA G D, RAMíREZ G S, GARCíAS, et al. A comparison on scalability for batch big data processing on Apache Spark and Apache Flink [J]. BigData Analytics, 2017, 2 (1): 1.

[8] ARDAGNO C A, BELLANDI V, BEZZI M, et al. Model-based big data analytics-as-a-service: take big data to the next level [J]. IEEE Transaction Serveices Computing, 2021, 14 (2): 516－529.

[9] RAGHAVENDRA K. The anatomy of big data computing [J]. Software: Practice and Experience, 2016, 46 (1): 79－105.

[10] JI C Q. Big data processing in cloud computing environments [C] // 2012 12th International Symposium on Pervasive Systems, Algorithms and Networks, IEEE, 2012, 16 (2): 69－75.

[11] MORELLI E A, MARK S S. Real-time dynamic modeling: data information requirements and flight-test results [J]. Journal of Aircraft, 2019, 46 (6): 1894－1905.

[12] KORSUN O N, POPLAVSKY B K, PRIHODKO S J. Intelligent support for aircraft flight test data processing in problem of engine thrust estimation [J]. Procedia Computer Science, 2017, 103 (1): 82－87.

[13] MUGTUSSIDS I B. Flight data processing techniques to identify unusual events [C] //Diss. Virginia Tech., 2000.

[14] LADWIG, GüNTER, TRAN T. Linked data query processing strategies [C] //ISWC, 2010.

[15] REN Y J. Data query mechanism based on hash computing power of blockchain in internet of things [J]. Sensors, 2019, 20 (1): 207.

[16] KOSSMANN, DONALD. The state of the art in distributed query processing [J]. ACM Computing Surveys (CSUR), 2000, 32 (4): 422－469.

[17] AMADEO, MARICA, CAMPOLO C, et al. Multi-source data retrieval in IoT via named data networking [C] // Proceedings of the 1st ACM Conference on Information-Centric Networking, 2014.

[18] AKERKAR, RAJENDRA. Big data computing [M]. Crc Press, 2013.

[19] ROSSMANN M G, CORNELIS G. BEEK V. Data processing [J]. Acta Crystallographica Section D: Biological Crystallography, 1999, 55 (10): 1631－1640.

[20] BERTHOLD, MICHAEL, HAND D J. Intelligent data analysis [M]. Berlin: Springer, 2003.