



(12) 发明专利申请

(10) 申请公布号 CN 115494810 A
(43) 申请公布日 2022.12.20

(21) 申请号 202211304149.X

(22) 申请日 2022.10.24

(71) 申请人 西北工业大学

地址 710072 陕西省西安市友谊西路127号

(72) 发明人 周琛淏 焦俊玲

(74) 专利代理机构 西安嘉思特知识产权代理事
务所(普通合伙) 61230

专利代理人 刘长春

(51) Int.Cl.

G05B 19/418 (2006.01)

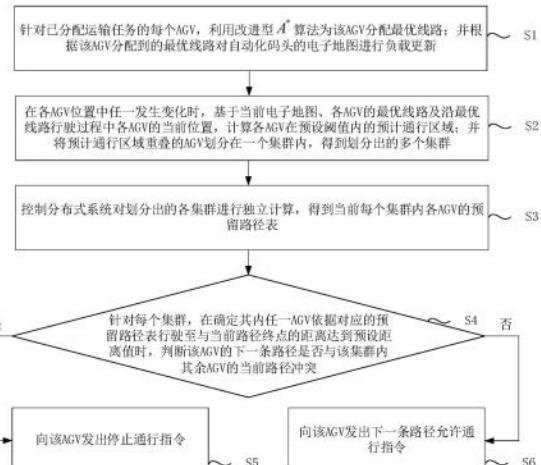
权利要求书5页 说明书23页 附图10页

(54) 发明名称

一种自动化码头的AGV在线路径规划方法

(57) 摘要

本发明公开了一种自动化码头的AGV在线路径规划方法，包括：对已分配运输任务的各AGV利用改进型A*算法分配最优线路并更新电子地图负载；在位置发生变化时，计算各AGV在预设阈值内的预计通行区域；将预计通行区域重叠的AGV划分在一个集群内；控制分布式系统对各集群独立计算得到各AGV的预留路径表；针对每个集群在确定其内任一AGV依据对应的预留路径表行驶至与当前路径终点的距离达到预设距离值时，判断下一条路径是否与集群内其余AGV的当前路径冲突；若是向该AGV发出停止通行指令；若否向该AGV发出下一条路径允许通行指令。本发明对自动化码头大规模AGV系统的路径规划能减少变道转弯和车辆拥堵，避免车辆冲突及死锁。



1.一种自动化码头的AGV在线路径规划方法,其特征在于,应用于自动化码头的中央控制系统,所述方法包括:

针对已分配运输任务的每个AGV,利用改进型A^{*}算法为该AGV分配最优线路;并根据该AGV分配到的最优线路对所述自动化码头的电子地图进行负载更新;其中,所述改进型A^{*}算法是基于当前电子地图中各边上的负载,以及添加有相邻两边的转弯时间成本、平衡负载成本和预防冲突死锁成本在内的评价函数,以边的评价函数值为决策实现最优线路计算;一条边上的负载表示依据各AGV分配到的最优线路,电子地图中该条边上已规划且尚未通过的AGV数量;

在各AGV位置中任一发生变化时,基于当前电子地图、各AGV的最优线路及沿最优线路行驶过程中各AGV的当前位置,计算各AGV在预设阈值内的预计通行区域;并将预计通行区域重叠的AGV划分在一个集群内,得到划分出的多个集群;

控制分布式系统对划分出的各集群进行独立计算,得到当前每个集群内各AGV的预留路径表;其中,每个AGV的预留路径表包括其预计通行区域对应的局部线路被划分出的多条路径,每条路径为一次连续行驶的最小边集合;

针对每个集群,在确定其内任一AGV依据对应的预留路径表行驶至与当前路径终点的距离达到预设距离值时,判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突;

若是,向该AGV发出停止通行指令;若否,向该AGV发出下一条路径允许通行指令;其中,所述中央控制系统在AGV行驶过程中接收各AGV对其最优线路中已通行边的反馈,并实时更新电子地图中的负载。

2.根据权利要求1所述的自动化码头的AGV在线路径规划方法,其特征在于,所述根据该AGV分配到的最优线路对所述自动化码头的电子地图进行负载更新,包括:

将所述自动化码头的电子地图中,该AGV分配到的最优线路涉及的边上的负载加一;其中,该AGV分配到的最优线路涉及的边包括通行边和冲突边;

相应的,所述中央控制系统在AGV行驶过程中接收各AGV对其最优线路中已通行边的反馈,并实时更新电子地图中的负载,包括:

所述中央控制系统针对接收到的每个已通行边,依据反馈该已通行边的AGV数量,减少当前电子地图中该已通行边及相关冲突边上的负载数量。

3.根据权利要求1或2所述的自动化码头的AGV在线路径规划方法,其特征在于,针对已分配运输任务的任一个AGV,利用改进型A^{*}算法为该AGV分配最优线路的过程,包括:

步骤a1,获取该AGV的运输任务的起点b和终点c,以及当前电子地图中各边上的负载;

步骤a2,针对首次迭代,获取为空集的OpenList、ClosedList,并设置起点b为首次迭代的父节点,执行步骤a3;

步骤a3,寻找以当前次迭代的父节点为起点的可达边集合A;

步骤a4,针对所述可达边集合A中每一条边a,利用预设的评价函数计算公式计算其当前的评价函数值F(a);

步骤a5,基于该条边a与当前的OpenList和ClosedList的所属关系,以及该条边a当前的F(a)与原评价函数值的比较关系,更新该条边a的List归属;

步骤a6,判断是否所述可达边集合A中每一条边a完成List归属更新;若否,执行步骤

a4;若是,执行步骤a7;

步骤a7,判断当前的OpenList是否为空;若是,执行步骤a8;若否,执行步骤a9;

步骤a8,确定针对该AGV的运输任务不存在最优线路;

步骤a9,将当前的OpenList中,评价函数值F(a)最小的一条边转入当前的ClosedList中,执行步骤a10;

步骤a10,判断当前的ClosedList中最后一条边a*的终点是否是终点c;若是,执行步骤a11;若否,执行步骤a12;

步骤a11;根据当前的ClosedList中记录的信息,确定一条从起点b到终点c的最短线路作为该AGV的最优线路;

步骤a12,将当前的ClosedList中最后一条边a*的终点作为下一次迭代的父节点,执行步骤a3。

4. 根据权利要求3所述的自动化码头的AGV在线路径规划方法,其特征在于,所述基于该条边a与当前的OpenList和ClosedList的所属关系,以及该条边a当前的F(a)与原评价函数值的比较关系,更新该条边a的List归属,包括:

若该条边a不在当前的OpenList和ClosedList中的任一个,将当前的(a,F(a))加入到当前的OpenList中;

若该条边a在当前的OpenList中,但不在当前的ClosedList中,判断当前的评价函数值F(a)是否小于当前的OpenList中该条边a的原评价函数值;若是,使用当前的(a,F(a))更新当前的OpenList中该条边a原有的信息;若否,维持当前的OpenList中该条边a原有的信息不变;

若该条边a在当前的ClosedList中,但不在当前的OpenList中,判断当前的评价函数值F(a)是否小于当前的ClosedList中该条边a的原评价函数值;若是,将当前的ClosedList中该条边a的信息删除,并将当前的(a,F(a))加入当前的OpenList中;若否,维持当前的ClosedList中该条边a原有的信息不变,以及维持当前的OpenList不变。

5. 根据权利要求3所述的自动化码头的AGV在线路径规划方法,其特征在于,所述预设的评价函数计算公式,包括:

$$F(e) = T^v(e) + T^r(e) + T^l(e) + T^c(e) + T^n(e)$$

其中,F(e)表示对检测边e计算得到的评价函数值;T^v(e)表示正常行驶的时间成本;T^r(e)表示相邻两边的转弯时间成本;T^l(e)表示平衡负载成本;T^c(e)表示预防冲突死锁成本;Tⁿ(e)表示距离任务终点的预计通行时间;其中:

$$T^v(e) = \sum_{e_i=e^0, \dots, e} t^v(e_i)$$

其中,e_i=e⁰,...e表示AGV沿着最优路线行驶中的各条边,e⁰为起始边,e为检测边;t^v(e_i)表示AGV在单条边e_i上的通行时长;

$$\begin{aligned} T^r(e) &= \sum_{e_i=e^0, \dots, e} t^r(e_{i-1}, e_i) \\ &= \sum_{e_i=e^0, \dots, e} \{B_1 \cdot \text{Abs}[\theta(e_i) - \theta(e_{i-1})] \cdot L\} \end{aligned}$$

其中,t^r(e_{i-1},e_i)表示相邻两边e_{i-1}和e_i之间的转弯时间成本;B₁表示转弯时间系数;θ

(e_{i-1}) 和 $\theta(e_i)$ 分别表示相邻两边与水平方向的弧度; $Abs[\theta(e_i) - \theta(e_{i-1})]$ 表示求该弧度的绝对值; L 表示转弯半径;

$$\begin{aligned} T^l(e) &= \sum_{e_i=e^0, \dots, e} t^l(e_i) \\ &= \sum_{e_i=e^0, \dots, e} [B_2 \cdot \sigma(e_i) \cdot t^v(e_i)] \end{aligned}$$

其中, $t^l(e_i)$ 表示每条边 e_i 的负载成本; B_2 表示拥堵时间系数; $\sigma(e_i)$ 表示边 e_i 上当前占用且尚未释放的负载量;

$$\begin{aligned} T^c(e) &= \sum_{e_i=e^0, \dots, e} t^c(e_i) \\ &= \sum_{e_i=e^0, \dots, e} \left\{ B_3 \cdot \sum_{\forall e_k \in Conflicts(e_i)} [\sigma(e_k) \cdot t^v(e_k)] \right\} \end{aligned}$$

其中, $t^c(e_i)$ 表示每条边 e_i 的预防冲突死锁成本; e_k 表示边 e_i 的冲突边; $Conflicts(e_i)$ 表示边 e_i 的冲突边集合; B_3 表示冲突时间系数;

$$T^n(e) = \frac{\sqrt{(x(t) - x_2(e))^2 + (y(t) - y_2(e))^2}}{\delta}$$

其中, δ 表示AGV匀速行驶速度; $x(t)$ 和 $y(t)$ 表示AGV任务结束边 t 的终点的横纵坐标; $x_2(e)$ 和 $y_2(e)$ 表示检测边 e 的终点的横纵坐标; AGV任务包括运输任务。

6. 根据权利要求1所述的自动化码头的AGV在线路径规划方法, 其特征在于, 所述预设阈值包括预设时间阈值或预设距离阈值; 其中, 所述预设距离阈值至少为AGV最短停车距离对应边数的10倍数; 所述AGV最短停车距离表示AGV以最大速度行驶时到完全停车所需的最短距离。

7. 根据权利要求1所述的自动化码头的AGV在线路径规划方法, 其特征在于, 所述将预计通行区域重叠的AGV划分在一个集群内, 得到划分出的多个集群, 包括:

在对当前所有AGV依次遍历过程中, 对遍历到的每个AGV, 分别判断该AGV与其余每个AGV的预计通行区域是否存在重复的边; 如果是, 将两者划分至同一集群, 否则, 将两者划分至各自对应的集群; 并在当前所有AGV依次遍历结束后得到划分出的多个集群。

8. 根据权利要求1或7所述的自动化码头的AGV在线路径规划方法, 其特征在于, 所述得到当前每个集群内各AGV的预留路径表, 包括:

针对当前任一集群, 确定该集群内所有AGV对应的优先级序列, 并按照所述优先级序列, 利用预留区域划分方法, 依次计算当前该集群内各AGV的预留路径表。

9. 根据权利要求8所述的自动化码头的AGV在线路径规划方法, 其特征在于, 针对任一AGV, 利用预留区域划分方法, 计算预留路径表的过程, 包括:

步骤b1, 获取该AGV在划分至所在集群时计算的预计通行区域所对应的目标边集合 $(e_1, e_2, \dots, e_{M-1}, e_M)$ 以及所在集群当前的彩色路径死锁检测图; 并为该AGV设置初始为空的预留路径表;

步骤b2, 针对该AGV的当前次预留, 获取当前次的预留区域, 并将所述当前次的预留区

域的前一条边作为预留边;其中,针对首次预留,将所述目标边集合中 e_M 作为预留区域,将前一条边 e_{M-1} 作为预留边;

步骤b3,使用彩色路径死锁检测方法检测该AGV当前次的预留请求与当前已确定的预留方案是否产生死锁情况;若否,执行步骤b4;若是,执行步骤b8;其中,该AGV当前次的预留请求由该AGV当前次预留确定的预留边和预留区域构成;所述当前已确定的预留方案包括当前该AGV所在集群中各AGV已确定的预留路径表;针对首个AGV的首次预留,当前已确定的预留方案为空;

步骤b4,将该AGV当前次的预留请求记录在该AGV的预留路径表和当前的彩色路径死锁检测图中;

步骤b5,判断所述目标边集合中的第一条边 e_1 是否已被确定为预留边记录在该AGV的预留路径表和当前的彩色路径死锁检测图中;若否,执行步骤b6;若是,执行步骤b7;

步骤b6,将该AGV当前次的预留请求中的预留边作为下一次预留中的预留区域,返回步骤b2;

步骤b7,停止计算,获得该AGV的预留路径表;

步骤b8,将当前次的预留边向前移动一条边,使得预留区域向前扩大一条边,更新当前次的预留请求,并执行步骤b3。

10.根据权利要求9所述的自动化码头的AGV在线路径规划方法,其特征在于,所述使用彩色路径死锁检测方法检测该AGV当前次的预留请求与当前已确定的预留方案是否产生死锁情况,包括:

步骤c1,针对该AGV当前次的预留请求(预留边q和预留区域S),将预留边q设为根节点,并作为首次迭代的父节点,执行步骤c2;

步骤c2,依次检查所有父节点查找到的可达子节点是否满足要求为:保持从根节点q到该子节点之间不包含颜色 C_{pj} 且无重复颜色的彩色路径;若是,执行步骤c3;若否,执行步骤c4,表示不存在从预留边q至预留区域S中的边 $s \in S$ 之间的一条彩色路径,则确定该AGV当前次的预留请求与当前已确定的预留方案未产生死锁情况;其中,在该AGV之前,对应的彩色路径死锁检测图中为已经完成轨迹规划的每个AGV的线路分别指定有一个不同颜色, C_{pj} 表示彩色路径死锁检测图中该AGV所在线路Pj的颜色;

步骤c3,如果查找到的可达子节点与其父节点的连通边的颜色不是 C_{pj} ,且不同于从根节点q到该可达子节点的父节点间的彩色路径中已有的其他颜色,判断该连通边是否是预留区域S中的一条边;若否,执行步骤c5,将该可达子节点作为下一次迭代的父节点,执行步骤c2;若是,执行步骤c6,找到一条从预留边q至预留区域S中的边 $s \in S$ 之间的一条彩色路径,则确定该AGV当前次的预留请求与当前已确定的预留方案产生死锁情况。

11.根据权利要求1或10所述的自动化码头的AGV在线路径规划方法,其特征在于,所述判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,包括:

判断该AGV的下一条路径对应的边集合中的每一条边,是否在该集群内其余任一AGV的当前路径对应的边集合中;

若是,判定该AGV的下一条路径与该集群内其余AGV的当前路径冲突,反之,判定该AGV的下一条路径与该集群内其余AGV的当前路径不冲突。

12.根据权利要求1所述的自动化码头的AGV在线路径规划方法,其特征在于,所述向该

AGV发出下一条路径停止通行指令之后,所述方法还包括:

响应于该AGV的定时请求,再次判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,并根据判断结果向该AGV发出相应的指令。

一种自动化码头的AGV在线路径规划方法

技术领域

[0001] 本发明属于AGV调度、AGV控制、路径规划、导航和智能交通领域，具体涉及一种自动化码头的AGV在线路径规划方法。

背景技术

[0002] 港口作为贸易交流与货物运输的枢纽，是海运的关键组成部分。目前，全世界港口中传统人工操作码头占集装箱码头的98%以上。近年来，随着海运集装箱业务量迅速增加，对港口装卸速度有了新的要求，自动化集装箱码头(简称为自动化码头)应运而生。

[0003] 相较于传统码头，自动化码头最突出的特点是岸边集装箱装卸、集装箱水平运输和堆场集装箱装卸等环节实现了全程智能化操作，从而提高了码头的工作效率，缓解了吞吐压力。大部分自动化港口均采用AGV(Automated Guided Vehicle，自动导引小车)作为水平运输载具，负责完成集装箱在岸边至堆场之间的搬运。AGV一般是通过中央控制系统控制其前进、转弯以及后退等行为，中央控制系统为AGV计算并分配最优线路，AGV则按照规划好的线路行驶。AGV作业的高效性和安全性对于提升码头吞吐量和竞争力至关重要。随着港口吞吐量持续增长和码头规模不断扩大，自动化码头需要部署更多的AGV以保证运输效率。

[0004] AGV的路径规划是整个水平运输作业过程中的核心问题。通常，为了便于对AGV进行精准控制，首先需要对自动化码头内部的AGV运输网络的物理环境进行建模，其目的是将环境中的物理信息抽象成算法能够处理的数据，搭建路径规划所用的电子地图，关键信息包括AGV可行驶车道、路网中的节点、车道的权重等；其中，AGV可行驶车道被抽象为边的集合，其交点为节点，也称为逻辑点。中小规模的自动化码头的水平运输网络可以由上万个逻辑点与逻辑点之间的各连通边构成。

[0005] 在电子地图基础上，AGV的路径规划首先进行线路规划，完成最优线路搜索，AGV线路规划是指根据某些优化目标(如运行时间最短、行驶线路最短、车辆行驶成本最小等)在物理环境抽象得到的电子地图中采用优化算法找到从起点至终点的一条最优解或接近最优解的线路，使得AGV后续按照找到的最优路线行驶。目前常用的线路规划方法主要包括基于图论的最短路算法，如广度优先搜索、深度优先搜索、Floyd算法、Dijkstra算法以及A^{*}算法等，还包括基于智能仿生学的启发式算法，如蚁群算法、粒子群算法和遗传算法等，以及基于人工智能的Q-Learning算法和神经网络算法等。但针对线路规划问题，自动化码头的决策过程需要具有高实时性，目前常用的线路规划方法大多无法在极短的时间内做出较优的决策，其中A^{*}算法虽然具有快速搜索能力，但其以距离最短或者时间最快为优化目标，难以从车辆实际行驶角度找到全局最优的规划方案；尽管也有一些其余算法在求解最优线路时考虑到了AGV变道与转弯对于运输效率的影响，但是并没有考虑大规模AGV车辆拥堵所导致的全局车辆运输效率低下的问题。

[0006] 在AGV获得最优线路之后，还需要进行无障碍通行方案的规划，这是由于在实际应用中，AGV的工作环境动态变化，通常难以不间断地从规划路线的起点行驶至终点，否则极可能发生冲突、死锁。因此，对于AGV的路径规划研究除了保障线路较优以外，还需要对系统

中多辆AGV的行驶过程进行协同调控以防止发生冲突或者陷入死锁状态。目前,常用的死锁预防方法有预留区域法、时间窗法、Petri网格法以及交通控制法等。其中,时间窗法、Petri网格法实现AGV行驶过程的死锁预防依赖于AGV行驶过程的精确预测,当AGV因故障、起步慢或者中途等待时间过长等原因导致与计划过程不一致时,需要重新完成整体的通行计划,否则很有可能导致冲突死锁情况,因此,常用于冲突死锁情况能够在短时间内轻易解决的小型仓储式搬运AGV,而不适用于大型集装箱运载AGV。交通控制法虽然能够解决常见的冲突情况,但是针对大规模AGV路径规划问题中多辆AGV线路重合的情况,常常使较多车辆等待较长的时间,容易陷入交通瘫痪。相比之下,基于预留区域划分的预留路径表法对于保障AGV行驶过程的安全性具有较佳的表现,但该算法通常计算复杂度非常高,计算时间随着AGV数量增加会呈指数型增长,因此,不适合对实时性要求极高的自动化码头水平运输系统。

[0007] 因此,针对实时性决策要求极高的自动化码头的大规模AGV系统,如何为各个AGV规划出最优线路以减少变道转弯、避免车辆拥堵等,并且避免AGV在行驶过程中出现冲突以及死锁情况,是本领域内一个亟待解决的问题。

发明内容

[0008] 为了解决现有技术中的上述问题,本发明实施例提供了一种自动化码头的AGV在线路径规划方法,应用于自动化码头的中央控制系统,具体技术方案如下:

[0009] 针对已分配运输任务的每个AGV,利用改进型A^{*}算法为该AGV分配最优线路;并根据该AGV分配到的最优线路对所述自动化码头的电子地图进行负载更新;其中,所述改进型A^{*}算法是基于当前电子地图中各边上的负载,以及添加有相邻两边的转弯时间成本、平衡负载成本和预防冲突死锁成本在内的评价函数,以边的评价函数值为决策实现最优线路计算;一条边上的负载表示依据各AGV分配到的最优线路,电子地图中该条边上已规划且尚未通过的AGV数量;

[0010] 在各AGV位置中任一发生变化时,基于当前电子地图、各AGV的最优线路及沿最优线路行驶过程中各AGV的当前位置,计算各AGV在预设阈值内的预计通行区域;并将预计通行区域重叠的AGV划分在一个集群内,得到划分出的多个集群;

[0011] 控制分布式系统对划分出的各集群进行独立计算,得到当前每个集群内各AGV的预留路径表;其中,每个AGV的预留路径表包括其预计通行区域对应的局部线路被划分出的多条路径,每条路径为一次连续行驶的最小边集合;

[0012] 针对每个集群,在确定其内任一AGV依据对应的预留路径表行驶至与当前路径终点的距离达到预设距离值时,判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突;

[0013] 若是,向该AGV发出停止通行指令;若否,向该AGV发出下一条路径允许通行指令;其中,所述中央控制系统在AGV行驶过程中接收各AGV对其最优线路中已通行边的反馈,并实时更新电子地图中的负载。

[0014] 在本发明的一个实施例中,所述根据该AGV分配到的最优线路对所述自动化码头的电子地图进行负载更新,包括:

[0015] 将所述自动化码头的电子地图中,该AGV分配到的最优线路涉及的边上的负载加

一；其中，该AGV分配到的最优线路涉及的边包括通行边和冲突边；

[0016] 相应的，所述中央控制系统在AGV行驶过程中接收各AGV对其最优线路中已通行边的反馈，并实时更新电子地图中的负载，包括：

[0017] 所述中央控制系统针对接收到的每个已通行边，依据反馈该已通行边的AGV数量，减少当前电子地图中该已通行边及相关冲突边上的负载数量。

[0018] 在本发明的一个实施例中，针对已分配运输任务的任一个AGV，利用改进型A^{*}算法为该AGV分配最优线路的过程，包括：

[0019] 步骤a1，获取该AGV的运输任务的起点b和终点c，以及当前电子地图中各边上的负载；

[0020] 步骤a2，针对首次迭代，获取为空集的OpenList、ClosedList，并设置起点b为首次迭代的父节点，执行步骤a3；

[0021] 步骤a3，寻找以当前次迭代的父节点为起点的可达边集合A；

[0022] 步骤a4，针对所述可达边集合A中每一条边a，利用预设的评价函数计算公式计算其当前的评价函数值F(a)；

[0023] 步骤a5，基于该条边a与当前的OpenList和ClosedList的所属关系，以及该条边a当前的F(a)与原评价函数值的比较关系，更新该条边a的List归属；

[0024] 步骤a6，判断是否所述可达边集合A中每一条边a完成List归属更新；若否，执行步骤a4；若是，执行步骤a7；

[0025] 步骤a7，判断当前的OpenList是否为空；若是，执行步骤a8；若否，执行步骤a9；

[0026] 步骤a8，确定针对该AGV的运输任务不存在最优线路；

[0027] 步骤a9，将当前的OpenList中，评价函数值F(a)最小的一条边转入当前的ClosedList中，执行步骤a10；

[0028] 步骤a10，判断当前的ClosedList中最后一条边a^{*}的终点是否是终点c；若是，执行步骤a11；若否，执行步骤a12；

[0029] 步骤a11；根据当前的ClosedList中记录的信息，确定一条从起点b到终点c的最短线路作为该AGV的最优线路；

[0030] 步骤a12，将当前的ClosedList中最后一条边a^{*}的终点作为下一次迭代的父节点，执行步骤a3。

[0031] 在本发明的一个实施例中，所述基于该条边a与当前的OpenList和ClosedList的所属关系，以及该条边a当前的F(a)与原评价函数值的比较关系，更新该条边a的List归属，包括：

[0032] 若该条边a不在当前的OpenList和ClosedList中的任一个，将当前的(a, F(a))加入到当前的OpenList中；

[0033] 若该条边a在当前的OpenList中，但不在当前的ClosedList中，判断当前的评价函数值F(a)是否小于当前的OpenList中该条边a的原评价函数值；若是，使用当前的(a, F(a))更新当前的OpenList中该条边a原有的信息；若否，维持当前的OpenList中该条边a原有的信息不变；

[0034] 若该条边a在当前的ClosedList中，但不在当前的OpenList中，判断当前的评价函数值F(a)是否小于当前的ClosedList中该条边a的原评价函数值；若是，将当前的

ClosedList中该条边a的信息删除，并将当前的(a,F(a))加入当前的OpenList中；若否，维持当前的ClosedList中该条边a原有的信息不变，以及维持当前的OpenList不变。

[0035] 在本发明的一个实施例中，所述预设的评价函数计算公式，包括：

$$[0036] F(e) = T^v(e) + T^r(e) + T^l(e) + T^c(e) + T^n(e)$$

[0037] 其中，F(e)表示对检测边e计算得到的评价函数值；T^v(e)表示正常行驶的时间成本；T^r(e)表示相邻两边的转弯时间成本；T^l(e)表示平衡负载成本；T^c(e)表示预防冲突死锁成本；Tⁿ(e)表示距离任务终点的预计通行时间；其中：

$$[0038] T^v(e) = \sum_{e_i=e^0, \dots, e} t^v(e_i)$$

[0039] 其中，e_i=e⁰,...e表示AGV沿着最优路线行驶中的各条边，e⁰为起始边，e为检测边；t^v(e_i)表示AGV在单条边e_i上的通行时长；

$$T^r(e) = \sum_{e_i=e^0, \dots, e} t^r(e_{i-1}, e_i)$$

$$[0040] = \sum_{e_i=e^0, \dots, e} \{B_1 \cdot \text{Abs}[\theta(e_i) - \theta(e_{i-1})] \cdot L\}$$

[0041] 其中，t^r(e_{i-1}, e_i)表示相邻两边e_{i-1}和e_i之间的转弯时间成本；B₁表示转弯时间系数；θ(e_{i-1})和θ(e_i)分别表示相邻两边与水平方向的弧度；Abs[θ(e_i) - θ(e_{i-1})]表示求该弧度的绝对值；L表示转弯半径；

$$[0042] T^l(e) = \sum_{e_i=e^0, \dots, e} t^l(e_i)$$

$$= \sum_{e_i=e^0, \dots, e} [B_2 \cdot \sigma(e_i) \cdot t^v(e_i)]$$

[0043] 其中，t^l(e_i)表示每条边e_i的负载成本；B₂表示拥堵时间系数；σ(e_i)表示边e_i上当前占用且尚未释放的负载量；

$$[0044] T^c(e) = \sum_{e_i=e^0, \dots, e} t^c(e_i)$$

$$= \sum_{e_i=e^0, \dots, e} \left\{ B_3 \cdot \sum_{\forall e_k \in \text{Conflicts}(e_i)} [\sigma(e_k) \cdot t^v(e_k)] \right\}$$

[0045] 其中，t^c(e_i)表示每条边e_i的预防冲突死锁成本；e_k表示边e_i的冲突边；Conflicts(e_i)表示边e_i的冲突边集合；B₃表示冲突时间系数；

$$[0046] T^n(e) = \frac{\sqrt{(x(t) - x_2(e))^2 + (y(t) - y_2(e))^2}}{\delta}$$

[0047] 其中，δ表示AGV匀速行驶速度；x(t)和y(t)表示AGV任务结束边t的终点的横纵坐标；x₂(e)和y₂(e)表示检测边e的终点的横纵坐标；AGV任务包括运输任务。

[0048] 在本发明的一个实施例中，所述预设阈值包括预设时间阈值或预设距离阈值；其中，所述预设距离阈值至少为AGV最短停车距离对应边数的10倍数；所述AGV最短停车距离

表示AGV以最大速度行驶时到完全停车所需的最短距离。

[0049] 在本发明的一个实施例中,所述将预计通行区域重叠的AGV划分在一个集群内,得到划分出的多个集群,包括:

[0050] 在对当前所有AGV依次遍历过程中,对遍历到的每个AGV,分别判断该AGV与其余每个AGV的预计通行区域是否存在重复的边;如果是,将两者划分至同一集群,否则,将两者划分至各自对应的集群;并在当前所有AGV依次遍历结束后得到划分出的多个集群。

[0051] 在本发明的一个实施例中,所述得到当前每个集群内各AGV的预留路径表,包括:

[0052] 针对当前任一集群,确定该集群内所有AGV对应的优先级序列,并按照所述优先级序列,利用预留区域划分方法,依次计算当前该集群内各AGV的预留路径表。

[0053] 在本发明的一个实施例中,针对任一AGV,利用预留区域划分方法,计算预留路径表的过程,包括:

[0054] 步骤b1,获取该AGV在划分至所在集群时计算的预计通行区域所对应的目标边集合($e_1, e_2, \dots, e_{M-1}, e_M$)以及所在集群当前的彩色路径死锁检测图;并为该AGV设置初始为空的预留路径表;

[0055] 步骤b2,针对该AGV的当前次预留,获取当前次的预留区域,并将所述当前次的预留区域的前一条边作为预留边;其中,针对首次预留,将所述目标边集合中 e_M 作为预留区域,将前一条边 e_{M-1} 作为预留边;

[0056] 步骤b3,使用彩色路径死锁检测方法检测该AGV当前次的预留请求与当前已确定的预留方案是否产生死锁情况;若否,执行步骤b4;若是,执行步骤b8;其中,该AGV当前次的预留请求由该AGV当前次预留确定的预留边和预留区域构成;所述当前已确定的预留方案包括当前该AGV所在集群中各AGV已确定的预留路径表;其中,针对首个AGV的首次预留,当前已确定的预留方案为空;

[0057] 步骤b4,将该AGV当前次的预留请求记录在该AGV的预留路径表和当前的彩色路径死锁检测图中;

[0058] 步骤b5,判断所述目标边集合中的第一条边 e_1 是否已被确定为预留边记录在该AGV的预留路径表和当前的彩色路径死锁检测图中;若否,执行步骤b6;若是,执行步骤b7;

[0059] 步骤b6,将该AGV当前次的预留请求中的预留边作为下一次预留中的预留区域,返回步骤b2;

[0060] 步骤b7,停止计算,获得该AGV的预留路径表;

[0061] 步骤b8,将当前次的预留边向前移动一条边,使得预留区域向前扩大一条边,更新当前次的预留请求,并执行步骤b3。

[0062] 在本发明的一个实施例中,所述使用彩色路径死锁检测方法检测该AGV当前次的预留请求与当前已确定的预留方案是否产生死锁情况,包括:

[0063] 步骤c1,针对该AGV当前次的预留请求(预留边q和预留区域S),将预留边q设为根节点,并作为首次迭代的父节点,执行步骤c2;

[0064] 步骤c2,依次检查所有父节点查找到的可达子节点是否满足要求为:保持从根节点q到该子节点之间不包含颜色 C_{pj} 且无重复颜色的彩色路径;若是,执行步骤c3;若否,执行步骤c4,表示不存在从预留边q至预留区域S中的边 $s \in S$ 之间的一条彩色路径,则确定该AGV当前次的预留请求与当前已确定的预留方案未产生死锁情况;其中,在该AGV之前,对应的

彩色路径死锁检测图中为已经完成轨迹规划的每个AGV的线路分别指定有一个不同颜色, C_{pj} 表示彩色路径死锁检测图中该AGV所在线路 P_j 的颜色;

[0065] 步骤c3,如果查找到的可达子节点与其父节点的连通边的颜色不是 C_{pj} ,且不同于从根节点q到该可达子节点的父节点间的彩色路径中已有的其他颜色,判断该连通边是否是预留区域S中的一条边;若否,执行步骤c5,将该可达子节点作为下一次迭代的父节点,执行步骤c2;若是,执行步骤c6,找到一条从预留边q至预留区域S中的边 $s \in S$ 之间的一条彩色路径,则确定该AGV当前次的预留请求与当前已确定的预留方案产生死锁情况。

[0066] 在本发明的一个实施例中,所述判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,包括:

[0067] 判断该AGV的下一条路径对应的边集合中的每一条边,是否在该集群内其余任一AGV的当前路径对应的边集合中;

[0068] 若是,判定该AGV的下一条路径与该集群内其余AGV的当前路径冲突,反之,判定该AGV的下一条路径与该集群内其余AGV的当前路径不冲突。

[0069] 在本发明的一个实施例中,所述向该AGV发出下一条路径停止通行指令之后,所述方法还包括:

[0070] 响应于该AGV的定时请求,再次判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,并根据判断结果向该AGV发出相应的指令。

[0071] 为有效改善自动化码头大规模AGV集群运输场景下面临的道路拥堵,甚至车辆冲突和死锁问题。本发明实施例提出了两阶段决策框架,设计了一种自动化码头的AGV在线路径规划方法,将原有路径规划问题拆解为在线线路规划与分布式轨迹规划两个紧密关联的子问题。在第一阶段,考虑水平运输路网中每段车道上的实时负载,以及相邻两边的转弯时间成本、平衡负载成本和预防冲突死锁成本,提出了改进型A*算法在线计算AGV执行运输等任务从起点至终点的最优线路,通过平衡路网中的车流,减小变道转弯,降低道路拥堵以及死锁的风险;在第二阶段,提出了分布式轨迹规划方法,在位置发生变动时,基于AGV实时状态,通过计算各AGV在预设阈值内的预计通行区域,将多个AGV动态地划分为多个相互独立的集群,并利用分布式计算方式,在集群内独立计算各AGV的预留路径表以规划AGV预留路径,在集群内的AGV沿着预留路径表的轨迹行驶至接近当前路径终点时,通过判断AGV的下一条路径是否与集群内其余AGV的当前路径冲突给出是否可以继续通行的指令,因而可以保证集群内的AGV无冲突无死锁通行。并且,第二阶段通过将AGV按实时空间位置拆分成多个集群,降低了参与轨迹规划的AGV数量以及路线长度,降低了计算复杂度;而且,由于集群之间在空间上距离较远,各个集群内部的轨迹规划不会影响其他AGV行驶,通过集群管理实现了分布式改进,可以将互相独立的计算任务部署在不同的计算资源上进行运算,提高了计算效率。

附图说明

[0072] 图1为本发明实施例中道路网络对应的赋权有向连通图的点、边与冲突边的示意图;

[0073] 图2为本发明实施例所提供的一种自动化码头的AGV在线路径规划方法的流程示意图;

- [0074] 图3为本发明实施例所提供的改进型A*算法的一种流程示意图；
- [0075] 图4(a)和图4(b)分别为本发明实施例某一时刻集群划分的一种示例；
- [0076] 图5为本发明实施例的预计通行区域的理解示意图；
- [0077] 图6为本发明实施例针对任一AGV，利用预留区域划分方法计算预留路径表的流程示意图；
- [0078] 图7为本发明实施例的彩色路径死锁检测示意图；
- [0079] 图8为本发明实施例的彩色路径死锁检测方法的流程示意图；
- [0080] 图9为本发明实施例用于说明预留区域划分方法的示例中的一个死锁示例图；
- [0081] 图10为本发明实施例用于说明预留区域划分方法的示例中的初始化死锁检测示意图；
- [0082] 图11(a)～图11(d)为本发明实施例用于说明预留区域划分方法的示例中的死锁检测图更新过程示意图；
- [0083] 图12为本发明实施例实验中大规模电子地图的示意图；
- [0084] 图13为本发明实施例实验中转弯时间因素对于单条线路直行边选择次数及转弯弧度均值的影响结果图；
- [0085] 图14为本发明实施例实验中负载的设置对单条线路平均负载与最大负载的影响结果图；
- [0086] 图15为本发明实施例实验中负载的设置对单条线路冲突边集平均负载与最大负载的影响结果图。

具体实施方式

[0087] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0088] 为了便于理解本发明实施例的方案，首先对本发明实施例涉及的场景问题和发明构思予以简介。

[0089] 现有的AGV路径规划问题可以根据研究目标分为两类，一类以寻找从起点到终点的最优线路为目标，且在搜索过程中不考虑车辆冲突、死锁等运行环节中可能发生的问题；另一类在保障无冲突且无死锁的条件下，以寻找从起点到终点的最优运行规划为目标。本发明实施例为了区分，采用线路规划和轨迹规划来命名这两类问题。

[0090] 自动化码头的真实环境中，道路网络可以抽象为赋权有向连通图 $G = (V, E)$ ，其中点 $v \in V$ 代表道路网络中车辆控制逻辑点，简称逻辑点，边 $e \in E$ 代表连通点与点之间的车辆行驶道路，简称连通边。一般为了实现AGV精细控制，逻辑点可以按照4米等间距以阵列的方式展开。考虑到车辆的实际行驶轨迹，本发明实施例中逻辑点之间的连通边可以是直线、曲线以及S弯等多种形式且长度已知。由于AGV物理尺寸较大且在转弯时车身会扫过一定区域，为了AGV运行安全并避免物理冲突，如图1所示，本发明实施例中道路网络对应的赋权有向连通图中每条边都定义了相应的冲突边集合 $\text{conflicts}(e)$ ，即当一辆AGV占用边 e 时，相关冲突边都会被同时占有以避免其他车辆驶入该区域，如图1所示，灰色矩形表示一个AGV，

加粗的边表示其冲突边。

[0091] 随着系统中运行的AGV数量不断增加,冲突边集合的存在使得AGV在执行运输等任务途中更容易发生车辆间冲突和死锁问题,从而影响水平运输作业的效率。常见的冲突可以通过AGV之间的简单避让或者停车等待快速得到解决,而循环等待的死锁情况则需要在规划AGV行驶轨迹的过程中提前处理,以避免死锁情况发生。针对以上可能出现的冲突及死锁问题,需要对AGV行驶过程进行合理的调度控制,使AGV能够安全、高效、有序地行驶至终点。

[0092] 发明人研究发现,随着码头规模扩大,将车流平衡地分散到多条线路上能够使AGV位置分散到更广的区域内,因而能够降低AGV之间在局部区域发生冲突和死锁的概率。因此确定决策系统将不再如传统方法一般从全局角度一次性设计AGV从起点至终点完整的行驶路径规划,而是考虑可以首先从干线线路规划角度,将车流分散到多条线路上以减少AGV在局部区域发生拥堵,之后从AGV运行轨迹角度,根据每辆AGV的实时位置与线路,为一定范围内有潜在冲突的AGV规划详细的行驶轨迹,确保AGV行进过程中无碰撞无死锁。在此构思基础上,为了针对实时性决策要求极高的自动化码头的大规模AGV系统,为各个AGV规划出最优线路以减少变道转弯、避免车辆拥堵等,并避免AGV在行驶过程中出现冲突以及死锁情况,本发明实施例在不牺牲作业效率与安全性的前提下,考虑从结构上降低算法复杂度以满足大规模水平运输系统的实时决策需求,最终基于AGV的实时运行状态设计了两阶段决策框架,将路径规划过程拆分为两个关联紧密的决策问题;(1)线路规划问题;(2)轨迹规划问题。对应提供了一种自动化码头的AGV在线路径规划方法,应用于自动化码头的中央控制系统。

[0093] 简单来说,本发明实施例的两阶段决策框架中,第一阶段以平衡车流为主要目标,在线更新网络负载并计算AGV最优线路,实现了一种线路规划方法;基于第一阶段的线路设计,第二阶段基于系统实时状态,动态地为AGV划分集群,并为集群内的AGV设计无冲突与死锁的运行轨迹,同时实现了多集群的分布式运算,实现了一种轨迹规划方法。

[0094] 以下对本发明实施例的两阶段决策框架所对应的自动化码头的AGV在线路径规划方法进行具体说明,如图2所示,该方法可以包括如下步骤:

[0095] S1,针对已分配运输任务的每个AGV,利用改进型A^{*}算法为该AGV分配最优线路;并根据该AGV分配到的最优线路对自动化码头的电子地图进行负载更新。

[0096] 本发明实施例中,针对自动化码头的AGV集群的一系列运输任务随着时间的推移,以在线的方式陆续产生,并由调度系统分配给指定的AGV,每个AGV陆续收到的运输任务将会排列组成其运输任务队列,其中每个运输任务包含任务起点、任务终点和任务产生时间等信息。任务起点表示AGV当前位置,任务起点和任务终点均与道路网络中的逻辑点对应。针对AGV的每个运输任务,均利用改进型A^{*}算法为AGV分配最优线路。当该AGV被分配最优线路后,将根据道路网络的结构与最优线路,按计划从任务起点行驶到任务终点,视为对应的运输任务结束。

[0097] 本发明实施例中,对自动化码头的每个AGV分配运输任务,可以利用现有技术实现,具体方法在此不做限制。同样,电子地图的建模方法也可以基于现有技术实现,在此不做说明。

[0098] 需要说明的是,本发明实施例中已分配运输任务的每个AGV所执行的任务并不仅

限于运输任务,还包括空车移动任务。可以理解的是,空车移动任务是为了从当前位置到达将要执行的运输任务的起点,以便于执行该运输任务,比如从当前位置A,空车移动至位置B,将位置B处的货物运输至位置C,以完成位置B至位置C的运输任务。因此,针对已分配运输任务的每个AGV,无论其要执行的任务是运输任务还是空车移动任务,均可以利用改进型A*算法为该AGV分配最优线路。

[0099] S1的目的是完成本发明实施例第一阶段的线路规划问题,根据运输等任务的起点和终点为AGV在赋权有向连通图中设计一条运行线路,优化目标是根据道路网络的实时状态,为AGV计算一条避免道路拥挤、减少变道转弯,且尽可能行驶时间最短的线路,以实现道路网络上的车流平衡。

[0100] 传统A*算法的原理是将到达终点的估计距离作为评价函数,通过搜索更少的节点信息找到一条相对较优的道路,提高搜索效率。然而传统A*算法以最短距离为优化目标,会导致车辆转弯次数过多以及交通堵塞的问题。

[0101] 本发明实施例提出了改进型A*算法。简要来说,改进型A*算法是基于当前电子地图中各边上的负载,以及添加有相邻两边的转弯时间成本、平衡负载成本和预防冲突死锁成本在内的评价函数,以边的评价函数值为决策实现最优线路计算。

[0102] 具体的,改进型A*算法以边为决策单位进行线路规划,首先根据车道长度、车辆行驶速度以及车辆制动性能等信息,计算AGV在各条边e上的正常通行时长、相邻车道上的转弯时间,并为赋权有向连通图中的边赋以相应权值,将传统A*算法原有的距离最短的优化目标转化为时间最短;其次,为实现分散车流的目的,本发明实施例引入了边上的实时负载的概念。其中,一条边上的负载表示依据各AGV分配到的最优线路,电子地图中该条边上已规划且尚未通过的AGV数量。并且,随着AGV被分配最优线路,以及在依据最优线路的实时运行过程中,本发明实施例不断更新边上的实时负载。具体来说,中央控制系统不仅根据该AGV分配到的最优线路对自动化码头的电子地图进行负载更新,而且,在AGV开始行驶后,中央控制系统在AGV行驶过程中接收各AGV对其最优线路中已通行边的反馈,并实时更新电子地图中的负载。关于负载更新的内容,将在后文中具体说明。

[0103] 以下以运输任务为例说明改进型A*算法。

[0104] 其中,相比传统A*算法,改进型A*算法对检测边e的评价函数进行了改进,添加有相邻两边的转弯时间成本、平衡负载成本和预防冲突死锁成本,所得到的预设的评价函数计算公式包括:

[0105] $F(e) = T^v(e) + T^r(e) + T^l(e) + T^c(e) + T^n(e)$

[0106] 其中,F(e)表示对检测边e计算得到的评价函数值; $T^v(e)$ 表示正常行驶的时间成本; $T^r(e)$ 表示相邻两边的转弯时间成本; $T^l(e)$ 表示平衡负载成本; $T^c(e)$ 表示预防冲突死锁成本; $T^n(e)$ 表示距离任务终点的预计通行时间。以下分别说明:

[0107] ①正常行驶的时间成本 $T^v(e)$:

$$[0108] T^v(e) = \sum_{e_i=e^0, \dots, e} t^v(e_i)$$

[0109] 其中,正常行驶的时间成本 $T^v(e)$ 表示从车辆当前任务的起始边到检测边的车辆正常行驶时间成本之和; $e_i = e^0, \dots, e$ 表示AGV沿着最优路线行驶中的各条边, e^0 为起始边,e为检测边; $t^v(e_i)$ 表示AGV在单条边 e_i 上的通行时长,表示单条边上的正常行驶时间成本,可

以由单条边长除以车辆行驶速度得到。

[0110] ②相邻两边的转弯时间成本 $T^r(e)$ ：

[0111] 相邻两边的转弯时间成本 $T^r(e)$ 表示从车辆当前任务的起始边到检测边的车辆转弯时间成本之和;AGV以正常速度通过各车道的时间固定且已知,车辆变道转弯时加减速行驶而产生的转弯时间与转弯角度和转弯半径有关,可由转弯弧长乘以转弯时间系数 B_1 确定。

$$\begin{aligned} [0112] T^r(e) &= \sum_{e_i=e^0, \dots, e} t^r(e_{i-1}, e_i) \\ &= \sum_{e_i=e^0, \dots, e} \{B_1 \cdot \text{Abs}[\theta(e_i) - \theta(e_{i-1})] \cdot L\} \end{aligned}$$

[0113] 其中, $t^r(e_{i-1}, e_i)$ 表示相邻两边 e_{i-1} 和 e_i 之间的转弯时间成本; B_1 表示转弯时间系数,根据情况合理选取数值; $\theta(e_{i-1})$ 和 $\theta(e_i)$ 分别表示相邻两边与水平方向的弧度; $\text{Abs}[\theta(e_i) - \theta(e_{i-1})]$ 表示求该弧度的绝对值; L 表示转弯半径,可以根据转弯场景设定,本发明实施例设置 L 为AGV车辆长度的四分之一。

[0114] 其中, $\theta(e_i)$ 和 $\theta(e_{i-1})$ 由以下公式确定:

$$[0115] \theta(e_i) = \arctan((y_2(e_i) - y_1(e_i)) / (x_2(e_i) - x_1(e_i))) * \pi / 180$$

[0116] 其中, $x_1(e_i)$ 、 $y_1(e_i)$ 表示边 e_i 起点的横、纵坐标, $x_2(e_i)$ 、 $y_2(e_i)$ 表示边 e_i 终点的横、纵坐标。

[0117] ③平衡负载成本 $T^l(e)$:

$$\begin{aligned} [0118] T^l(e) &= \sum_{e_i=e^0, \dots, e} t^l(e_i) \\ &= \sum_{e_i=e^0, \dots, e} [B_2 \cdot \sigma(e_i) \cdot t^v(e_i)] \end{aligned}$$

[0119] 其中,平衡负载成本 $T^l(e)$ 表示从车辆当前任务的起始边到检测边之间的各条边负载成本之和;

[0120] $t^l(e_i)$ 表示每条边 e_i 的负载成本,由该条边上的实时负载和通行时长 $t^v(e_i)$ 决定; B_2 表示拥堵时间系数,根据情况合理选取数值; $\sigma(e_i)$ 表示边 e_i 上当前占用且尚未释放的负载量,即该条边上的实时负载,随着更多线路需要通行同一条边,每条边更容易发生排队和拥堵现象,通行时长 $t^v(e_i)$ 将会显著增加。

[0121] ④预防冲突死锁成本 $T^c(e)$:

$$\begin{aligned} [0122] T^c(e) &= \sum_{e_i=e^0, \dots, e} t^c(e_i) \\ &= \sum_{e_i=e^0, \dots, e} \left\{ B_3 \cdot \sum_{\forall e_k \in \text{Conflicts}(e_i)} [\sigma(e_k) \cdot t^v(e_k)] \right\} \end{aligned}$$

[0123] 其中,预防冲突死锁成本 $T^c(e)$ 表示从车辆当前任务的起始边到检测边之间的各条边对应的预防冲突死锁成本之和。 $t^c(e_i)$ 表示每条边 e_i 的预防冲突死锁成本,由该条边对应的冲突边的实时负载与正常通行时长决定; e_k 表示边 e_i 的冲突边; $\text{Conflicts}(e_i)$ 表示边

e_i 的冲突边集合; B_3 表示冲突时间系数,根据情况合理选取数值。

[0124] ⑤距离任务终点的预计通行时间 $T^n(e)$:

[0125] 由于AGV在行驶过程中可以在逻辑点间灵活行驶,其移动不局限于前、后、左、右、左前、左后、右前、右后八个方向。可选的一种实施方式中,选用欧式距离计算当前检测边e到任务结束边t之间的预计通行时间:

$$[0126] T^n(e) = \frac{\sqrt{(x(t) - x_2(e))^2 + (y(t) - y_2(e))^2}}{\delta}$$

[0127] 其中, δ 表示AGV匀速行驶速度; $x(t)$ 和 $y(t)$ 表示AGV任务结束边t的终点的横纵坐标;

[0128] $x_2(e)$ 和 $y_2(e)$ 表示检测边e的终点的横纵坐标; AGV任务包括运输任务。公式中涉及的i、k均为自然数。

[0129] 以运输任务为例说明,请参见图3,可选的一种实施方式中,针对已分配运输任务的任一个AGV,利用改进型A*算法为该AGV分配最优线路的过程,可以包括以下步骤:

[0130] 步骤a1,获取该AGV的运输任务的起点b和终点c,以及当前电子地图中各边上的负载;

[0131] 步骤a2,针对首次迭代,获取为空集的OpenList、ClosedList,并设置起点b为首次迭代的父节点,执行步骤a3;

[0132] 其中,OpenList、ClosedList表示维护的两个列表,具体概念请结合传统A*算法理解,在此不做详细说明。

[0133] 步骤a3,寻找以当前次迭代的父节点为起点的可达边集合A;

[0134] 针对首次迭代,“当前次迭代的父节点”为起点b;针对首次迭代之后各次迭代时的“当前次迭代的父节点”,请参见后续步骤a12的相关说明。

[0135] 可达边表示与起点连通的边,具体概念请参见相关技术理解。

[0136] 步骤a4,针对可达边集合A中每一条边a,利用预设的评价函数计算公式计算其当前的评价函数值F(a);

[0137] 其中,预设的评价函数计算公式请参见前文相关描述。

[0138] 步骤a5,基于该条边a与当前的OpenList和ClosedList的所属关系,以及该条边a当前的F(a)与原评价函数值的比较关系,更新该条边a的List归属;

[0139] 可选的一种实施方式中,步骤a5包括:

[0140] 1) 若该条边a不在当前的OpenList和ClosedList中的任一个,将当前的(a, F(a))加入到当前的OpenList中。

[0141] 具体的,可以先判断该条边a是否不在当前的OpenList和ClosedList中的任一个;如果是,则将当前的(a, F(a))加入到当前的OpenList中,此时步骤a5结束。如果否,则继续判断该条边a是否在当前的OpenList中,但不在当前的ClosedList中。

[0142] 2) 若该条边a在当前的OpenList中,但不在当前的ClosedList中,判断当前的评价函数值F(a)是否小于当前的OpenList中该条边a的原评价函数值;若是,使用当前的(a, F(a))更新当前的OpenList中该条边a原有的信息;若否,维持当前的OpenList中该条边a原有的信息不变。

[0143] 具体的,判断该条边a是否在当前的OpenList中,但不在当前的ClosedList中,如果否,则说明该条边a在当前的ClosedList中,但不在当前的OpenList中,则按照3)的方式执行。如果该条边a在当前的OpenList中,但不在当前的ClosedList中,继续判断当前的评价函数值F(a)是否小于当前的OpenList中该条边a的原评价函数值;若是,使用当前的(a,F(a))更新当前的OpenList中该条边a原有的信息;此时步骤a5结束。若否,维持当前的OpenList中该条边a原有的信息不变;此时步骤a5结束。

[0144] 3)若该条边a在当前的ClosedList中,但不在当前的OpenList中,判断当前的评价函数值F(a)是否小于当前的ClosedList中该条边a的原评价函数值;若是,将当前的ClosedList中该条边a的信息删除,并将当前的(a,F(a))加入当前的OpenList中;若否,维持当前的ClosedList中该条边a原有的信息不变,以及维持当前的OpenList不变。

[0145] 具体的,该种情况下两种判断结果按照3)的对应方式执行完成后,此时步骤a5结束。

[0146] 步骤a6,判断是否可达边集合A中每一条边a完成List归属更新;若否,执行步骤a4;若是,执行步骤a7;

[0147] 具体的,步骤a4~步骤a5是遍历可达边集合A中每一条边a完成List归属更新,直至全部遍历完成后执行步骤a7。

[0148] 步骤a7,判断当前的OpenList是否为空;若是,执行步骤a8;若否,执行步骤a9;

[0149] 步骤a8,确定针对该AGV的运输任务不存在最优线路;

[0150] 步骤a9,将当前的OpenList中,评价函数值F(a)最小的一条边转入当前的ClosedList中,执行步骤a10;

[0151] 步骤a10,判断当前的ClosedList中最后一条边a*的终点是否是终点c;若是,执行步骤a11;若否,执行步骤a12;

[0152] 步骤a11;根据当前的ClosedList中记录的信息,确定一条从起点b到终点c的最短线路作为该AGV的最优线路;

[0153] 步骤a12,将当前的ClosedList中最后一条边a*的终点作为下一次迭代的父节点,执行步骤a3。

[0154] 本发明实施例在AGV线路规划中,通过考虑水平运输路网车道上的动态变化的实时负载,并在传统A*算法的行驶时间最短的优化目标中加入平衡路网中的车流、减少变道转弯等因素,以及考虑冲突边上车流的影响,能够在线路规划阶段降低AGV集群发生冲突和死锁的概率,为AGV计算一条避免道路拥挤、减少变道转弯,且尽可能行驶时间最短的最优线路。

[0155] 以下对负载更新的内容进行集中说明。

[0156] 针对每个AGV,当该AGV被分配最优线路后,根据该AGV分配到的最优线路对自动化码头的电子地图进行负载更新,包括:

[0157] 将自动化码头的电子地图中,该AGV分配到的最优线路涉及的边上的负载加一。

[0158] 其中,该AGV分配到的最优线路涉及的边包括通行边和冲突边。通行边即AGV会依次通行的边;冲突边即由于AGV通行带来物理限制的相邻边。

[0159] 如前,本发明实施例为实现分散车流的目的,引入了实时负载的概念,记录各个时刻道路上已经规划且尚未通过的AGV数量。针对每一个AGV,当使用改进型A*算法规划出最

优线路后,在该最优线路涉及的各边上实现负载加一。

[0160] 由于后续AGV需要依据分配的最优线路行驶,为了后续准确对各AGV进行线路规划和轨迹规划,除了最优线路分配后的负载更新,还需要在AGV行驶过程中进行实时负载更新。相应的,中央控制系统在AGV行驶过程中接收各AGV对其最优线路中已通行边的反馈,并实时更新电子地图中的负载,包括:

[0161] 中央控制系统针对接收到的每个已通行边,依据反馈该已通行边的AGV数量,减少当前电子地图中该已通行边及相关冲突边上的负载数量。

[0162] 也就是说,在AGV依据最优线路行驶过程中,当AGV通过并离开一条边,会将已通行边反馈给中央控制系统,中央控制系统会将电子地图中该已通行边及其相关冲突边上的负载减一,以释放对应边上的负载。

[0163] 可选的一种实施方式中,各AGV可以在通过并离开一条边时立即向中央控制系统反馈;当然,各AGV也可以依据固定的频率,比如每隔5秒等,向中央控制系统反馈一段时间内的已通行边,这都是合理的。

[0164] S2,在各AGV位置中任一发生变化时,基于当前电子地图、各AGV的最优线路及沿最优线路行驶过程中各AGV的当前位置,计算各AGV在预设阈值内的预计通行区域;并将预计通行区域重叠的AGV划分在一个集群内,得到划分出的多个集群;

[0165] 为了便于理解本发明实施例S2~S6对应的第二阶段的轨迹规划方法,首先对该阶段的发明构思和方案进行简要说明。

[0166] AGV收到中央控制系统为其分配的最优线路后,将沿着其最优线路行驶,则其将要通行的边以及对应的冲突边是已知的。从最优线路的静态角度考虑,各AGV的运行线路之间可能存在较多的重叠与冲突。传统的集中式死锁预防算法即是从最优线路整体出发,对系统中所有AGV的全线路均进行防冲突防死锁的方案计算。但由于自动化码头水平运输区域较大,该种方式不仅存在计算复杂度高、计算时间长的问题,而且经过长时间行驶后,AGV往往容易偏离计划。而发明人分析发现,在实际运行中,许多AGV之间可能相距较远,也就意味着这些AGV之间在短期内不会存在边的重叠与冲突,因此从空间层面来看,针对距离较远的AGV计算死锁预防的通行方案并无意义,因而发明人基于此结论设计了第二阶段的轨迹规划方法。具体是在AGV行驶过程中,根据系统实时状态不断为AGV设计行驶轨迹。为了提升系统的决策效率,先提出了动态集群划分方法,将系统中的所有AGV在时空范围上划分为多个互相独立的集群;再对每个集群内的AGV利用分布式计算方式独立计算无死锁的轨迹规划方案,具体是利用预留区域划分方法和彩色路径死锁检测方法实现。不仅能够减少计算复杂度,降低计算时长,还能够在局部区域实现更加精确的决策。与第一阶段的线路规划不同,第二阶段的轨迹规划是在已知线路上按逻辑边划分通行预留区域,使得AGV按照通行预留区域依指令分段行驶,从而确保各AGV在行驶过程中不会出现冲突和死锁情况。

[0167] S2中,本发明实施例引入了预计通行区域的概念,具体根据AGV在位置变动时刻的当前位置,预测表示未来的预设阈值内AGV的通行区域,如果有AGV的通行区域存在重叠,说明这些AGV在未来行驶过程中可能形成冲突和死锁,因而将预计通行区域存在重叠的AGV划为一个集群。

[0168] 可选的一种实施方式中,预设阈值包括预设时间阈值或预设距离阈值。

[0169] 其中,预设时间阈值可以根据需求选择,比如可以为1分钟、5分钟等。可选的一种

实施方式中,预设距离阈值至少为AGV最短停车距离对应边数的10倍数;当然,本发明实施例中,预设距离阈值为AGV最短停车距离对应边数的倍数并不局限于10,可以根据需求合理设置。AGV最短停车距离表示AGV以最大速度行驶时到完全停车所需的最短距离。可以理解的是,最短停车距离表示为一个长度,由于本发明实施例的道路网络中各边的长度已知,最短停车距离可以以边数表征,比如5条边等,因此,预设距离阈值可以表征边数。

[0170] 在各AGV位置中任一发生变化时,基于当前电子地图、各AGV的最优线路及沿最优线路行驶过程中各AGV的当前位置,计算各AGV在预设阈值内的预计通行区域,具体是针对每个AGV,根据该AGV在该时刻的位置、最优线路表征的边集,计算预设阈值内该AGV未来行驶区域所涉及的边集,得到预计通行区域。因此,预计通行区域表征为一个边集,其不仅包括通行边,也包括对应的冲突边。

[0171] 关于预计通行区域的确定过程,比如针对预设时间阈值,可以根据该AGV的行驶速度,计算其在预设时间阈值内可以行驶的长度;并根据道路网络中各边已知长度,计算其在预设时间阈值内可以行驶的边数。再根据该AGV在该时刻的位置和最优线路表征的边集,确定该AGV在该时刻起,未来预设时间阈值内将要行驶到的各个边以及相应的冲突边,从而得到这些边的集合构成预计通行区域。又比如针对预设距离阈值,由于其表征为一个边数,相比于预设时间阈值,可以直接根据该AGV在该时刻的位置和最优线路表征的边集,确定该AGV在该时刻起,预设距离阈值内将要行驶到的各个边以及相应的冲突边,从而得到这些边的集合构成预计通行区域。

[0172] 在得到某个时刻所有AGV的预计通行区域后,可选的一种实施方式中,将预计通行区域重叠的AGV划分在一个集群内,得到划分出的多个集群,包括:

[0173] 在对当前所有AGV依次遍历过程中,对遍历到的每个AGV,分别判断该AGV与其余每个AGV的预计通行区域是否存在重复的边;如果是,将两者划分至同一集群,否则,将两者划分至各自对应的集群;并在当前所有AGV依次遍历结束后得到划分出的多个集群。

[0174] 具体的,集群划分过程可以为:对于系统中正在运行的AGV,从*i*=1开始,检测第*i*个AGV与其余每个AGV的预计通行区域是否存在重复的边,将与第*i*个AGV的预计通行区域存在重复的边的AGV加入第*i*个AGV所在的集群;然后*i*加一后继续重复上述过程,直到所有AGV均已划分至确定的集群,则得到划分出的多个集群。其中,*i*为大于0的自然数。划分出的每个集群,可以以车辆列表的形式表征该集群内的所有AGV。

[0175] 请参见图4(a)和图4(b)给出的某一时刻集群划分的两种示例图进行理解。其中,图4(a)和图4(b)中各AGV的预计通行区域以点和线连接构成的折线表示,灰色线条表示预计通行区域,预计通行区域存在重复的边则为折线存在相交部分。针对图4(a)对应的时刻,由于其中的AGV1和AGV2的预计通行区域存在重复的边,因而两者被划分在集群1中,而由于AGV3的预计通行区域和AGV1~AGV2均不存在重复的边,因而AGV3被划分在单独的集群2中。针对图4(b)对应的时刻,由于其中的AGV1~AGV3的预计通行区域均存在重复的边,因而三者被划分在同一个集群1中。

[0176] 可以理解的是,本发明实施例中集群划分是动态进行的,受AGV位置变化触发,即使在划分集群后,只要AGV再次发生位置变动,仍需要再次进行集群划分。同时,可以理解的是,随着AGV的行驶,针对不同时刻,电子地图中的负载以及AGV的预计通行区域都在实时变化。

[0177] S3,控制分布式系统对划分出的各集群进行独立计算,得到当前每个集群内各AGV的预留路径表。

[0178] 在每次集群划分后,需要对每个集群的内部成员计算行驶轨迹,本发明实施例将互相独立的集群计算任务,利用分布式系统,部署给不同的处理器,这样,通过分布式计算可以缩短决策时间,提高系统的决策效率。

[0179] 其中,每个AGV的预留路径表包括其预计通行区域对应的局部线路被划分出的多条路径,每条路径为一次连续行驶的最小边集合。

[0180] 由于一个AGV的预计通行区域表示的是计算时刻,该AGV在预设阈值内未来行驶涉及的边的集合。可以理解的是,相对该AGV完整的最优线路而言,预计通行区域是一个局部线路,包含有道路网络的多个边以及相应的冲突边。预留路径表表示的是,将预计通行区域包含的多个边划分为多个段,即为多个预留区域,每一段表示一条路径,每条路径为一次连续行驶的最小边集合,一次连续行驶意味着AGV可以不停留地正常行驶。一条路径可能包含道路网络中的一条边或者多条边,需要根据各AGV的情况具体计算得到。关于AGV的预计通行区域可以参见图5理解,其中,矩形表示AGV,每个AGV的轨迹规划为多个预计通行区域,以斜线分割。

[0181] 可选的一种实施方式中,得到当前每个集群内各AGV的预留路径表,包括:

[0182] 针对当前任一集群,确定该集群内所有AGV对应的优先级序列,并按照优先级序列,利用预留区域划分方法,依次计算当前该集群内各AGV的预留路径表。

[0183] 其中,在需要执行预留区域划分的时刻,按照优先级序列的顺序,从第一个AGV至最后一个AGV依次执行预留区域划分方法得到各AGV的预留路径表。确定该集群内所有AGV对应的优先级序列的依据可以是表征AGV运输任务的紧急程度或者重要程度等的属性信息,比如,依据紧急程度由高到低将该集群内所有AGV排序得到一个优先级序列。当然,确定优先级序列的方式可以不限于以上所示。

[0184] 本发明实施例针对独立的集群,基于预留路径表提出了考虑集群空间范围与成员动态变化的局部无死锁的轨迹规划方法。该方法的主要思路是针对集群中每个AGV的线路依次执行预留区域划分方法,并将每一组已确定的AGV预留信息记录在死锁检测图上,在后续的计算过程中,通过彩色路径死锁检测方法判断新的预留请求是否与已确定的预留信息构成死锁,如果构成死锁则更改预留请求,直至无死锁时,确认该组预留请求并记录在死锁检测图上。

[0185] 可选的一种实施方式中,请参见图6,针对任一AGV,利用预留区域划分方法,计算预留路径表的过程,包括:

[0186] 步骤b1,获取该AGV在划分至所在集群时计算的预计通行区域所对应的目标边集合($e_1, e_2, \dots, e_{M-1}, e_M$)以及所在集群当前的彩色路径死锁检测图;并为该AGV设置初始为空的预留路径表;

[0187] 其中,集群当前的彩色路径死锁检测图上可能已经记录有集群其余AGV的预留请求记录。为了便于理解方案,在此对彩色路径死锁检测方法以及彩色路径死锁检测图进行简单介绍。

[0188] 彩色路径死锁检测方法主要作用是避免在预留路径时产生潜在死锁。在彩色路径死锁检测图 $G_D = (V_D, E_D)$ 中,节点 $v_D \in V_D$ 表示有向图G中的边 $e \in E$,边 e_D 表示有向图G中两条边

的关系。假设AGV当前正在边 $e_1 \in E$ 上,且需要预留边 $e_2 \in E$,称 $e_p = (e_1, e_2) \in E_p$ 存在,而反映在 G_p 上时一条具有代表该AGV特定颜色的边,即每一种颜色代表一条AGV线路的预留请求。如图7所示的彩色路径死锁检测示意图,彩色路径是指死锁检测图 G_p 上存在一条路径,其每条边的颜色各异,由于每种颜色代表一个特定的AGV的预留请求,一个彩色的环形路径表示一组AGV正在循环请求预留相应的边。在最差的情况下,相应的AGV恰好位于对应颜色的边上,每辆AGV都在请求预留已经被其他AGV占用的边,进而构成死锁。为了避免死锁产生,在预留区域划分决策时检测一组新的预留请求是否与已经确认的预留请求构成彩色的环,即可判断该AGV是否和其他AGV构成死锁。如果将要发生死锁,当前预留请求将不可行,需要通过改变预留请求的长度调整预留路径表,从而实现无死锁的预留区域划分。

[0189] 步骤b2,针对该AGV的当前次预留,获取当前次的预留区域,并将当前次的预留区域的前一条边作为预留边;

[0190] 其中,针对首次预留,将目标边集合中 e_M 作为预留区域,将前一条边 e_{M-1} 作为预留边。

[0191] 步骤b3,使用彩色路径死锁检测方法检测该AGV当前次的预留请求与当前已确定的预留方案是否产生死锁情况;若否,执行步骤b4;若是,执行步骤b8;

[0192] 其中,该AGV当前次的预留请求由该AGV当前次预留确定的预留边和预留区域构成;当前已确定的预留方案包括当前该AGV所在集群中各AGV已确定的预留路径表;针对首次预留,当前已确定的预留方案为空。

[0193] 步骤b4,将该AGV当前次的预留请求记录在该AGV的预留路径表和当前的彩色路径死锁检测图中;

[0194] 步骤b5,判断目标边集合中的第一条边 e_1 是否已被确定为预留边记录在该AGV的预留路径表和当前的彩色路径死锁检测图中;若否,执行步骤b6;若是,执行步骤b7;

[0195] 步骤b6,将该AGV当前次的预留请求中的预留边作为下一次预留中的预留区域,返回步骤b2;

[0196] 步骤b7,停止计算,获得该AGV的预留路径表;

[0197] 步骤b8,将当前次的预留边向前移动一条边,使得预留区域向前扩大一条边,更新当前次的预留请求,并执行步骤b3。

[0198] 需要说明的是,针对步骤b8,假设当前次的预留边为 e_x ,向前移动一条边使得预留区域向前扩大一条边意味着从边 e_x 向前移动至边 e_{x-1} 。

[0199] 其中,参见图8,针对步骤b3,使用彩色路径死锁检测方法检测该AGV当前次的预留请求与当前已确定的预留方案是否产生死锁情况,包括:

[0200] 步骤c1,针对该AGV当前次的预留请求(预留边q和预留区域S),将预留边q设为根节点,并作为首次迭代的父节点,执行步骤c2;

[0201] 步骤c2,依次检查所有父节点查找到的可达子节点,是否存在某个子节点满足以下要求:存在一条保持从根节点q到该子节点之间不包含颜色 C_{pj} 且无重复颜色的彩色路径;若是,执行步骤c3;若否,执行步骤c4,表示不存在从预留边q至预留区域S中的边 $s \in S$ 之间的一条彩色路径,则确定该AGV当前次的预留请求与当前已确定的预留方案未产生死锁情况;其中,在该AGV之前,对应的彩色路径死锁检测图中为已经完成轨迹规划的每个AGV的线路分别指定有一个不同颜色, C_{pj} 表示彩色路径死锁检测图中该AGV所在线路Pj的颜色;

[0202] 步骤c3,如果查找到的可达子节点与其父节点的连通边的颜色不是 C_{pj} ,且不同于从根节点q到该可达子节点的父节点间的彩色路径中已有的其他颜色,判断该连通边是否是预留区域S中的一条边;若否,执行步骤c5,将该可达子节点作为下一次迭代的父节点,执行步骤c2;若是,执行步骤c6,找到一条从预留边q至预留区域S中的边 $s \in S$ 之间的一条彩色路径,则确定该AGV当前次的预留请求与当前已确定的预留方案产生死锁情况。

[0203] 以下对利用预留区域划分方法,计算当前该集群内各AGV的预留路径表的过程举例说明。如图9所示,AGV1的线路为 $P_1 = (e_2, e_5)$, $P_2 = (e_5, e_7)$, $P_3 = (e_7, e_4)$, $P_4 = (e_9, e_4, e_2)$,假定用红色代表AGV1,橙色代表AGV2,黄色代表AGV3,绿色代表AGV4。以这种常见的四个AGV容易陷入循环等待的情况为例,在暂不考虑图中冲突边关系的情况下,按照AGV编号从小到大的顺序检测死锁、预防死锁并划分预留区域的步骤可以包括:

[0204] ①初始化彩色路径死锁检测图 $G_D = (V_D, E_D)$,如图10所示。

[0205] ②为运输车辆1计算预留路径表,尝试使 e_2 预留 e_5 ,即当前次的预留请求中预留区域为 e_5 ,预留边为 e_2 ;检查 G_D 中预留边 e_2 反映的节点 v_{De_2} ,至 G_D 中预留区域 e_5 反映的节点 v_{De_5} 之间,是否存在没有使用红色的一条彩色路径,显然没有,则确认 P_1 的预留信息(即 e_2 作为预留的请求边,预留 e_5 作为预留区域) $R_{P_1(e_2)} = e_5$,使用 $e_D = ((V_{e_2}, V_{e_5}), \text{ 红色})$ 来表示该组预留请求并添加至死锁检测图 G_D 上, G_D 更新为图11(a);

[0206] ③为运输车辆2计算预留路径表,尝试使 e_5 预留 e_7 ,即当前次的预留请求中预留区域为 e_7 ,预留边为 e_5 ;检查 G_D 中预留边 e_5 反映的节点 v_{De_5} ,至 G_D 中预留区域 e_7 反映的节点 v_{De_7} 之间,是否存在没有使用橙色的一条彩色路径,显然没有,则确认 P_2 的预留信息(即 e_5 作为预留的请求边,预留 e_7 作为预留区域) $R_{P_2(e_5)} = e_7$,使用 $e_D = ((V_{e_5}, V_{e_7}), \text{ 橙色})$ 来表示该组预留请求并添加至 G_D 上, G_D 更新为图11(b);

[0207] ④为运输车辆3计算预留路径表,尝试使 e_7 预留 e_4 ,即当前次的预留请求中预留区域为 e_4 ,预留边为 e_7 ;检查 G_D 中预留边 e_7 反映的节点 v_{De_7} ,至 G_D 中预留区域 e_4 反映的节点 v_{De_4} 之间,是否存在没有使用黄色的一条彩色路径,显然没有,则确认 P_3 的预留信息(即 e_7 作为预留的请求边,预留 e_4 作为预留区域) $R_{P_3(e_7)} = e_4$,使用 $e_D = ((V_{e_7}, V_{e_4}), \text{ 黄色})$ 来表示该组预留请求并添加至 G_D 上, G_D 更新为图11(c);

[0208] ⑤为运输车辆4计算预留路径表,尝试使 e_4 预留 e_2 ,即当前次的预留请求中预留区域为 e_2 ,预留边为 e_4 ;检查 G_D 中预留边 e_4 反映的节点 v_{De_4} ,至 G_D 中预留区域 e_2 反映的节点 v_{De_2} 之间,是否存在没有使用绿色的一条彩色路径,显然 $e_2 - e_5 - e_7 - e_4$ 是一条没有绿色的彩色路径,因此一旦使 e_4 预留 e_2 ,会导致四个运输车辆陷入循环等待的死锁状态,需将 e_4 沿 P_4 向上移动,使 e_9 预留 e_2 及 e_4 ,即修改后,当前次的预留请求中预留区域为 e_2 及 e_4 ,预留边为 e_9 ;检查 G_D 中预留边 e_9 反映的节点 v_{De_9} 至 v_{De_2} 之间以及 v_{De_9} 至 v_{De_4} 之间是否存在彩色路径,显然不存在,则确认现在 P_4 的预留信息 $R_{P_4(e_9)} = e_2 \cup e_4$,使用 $e_D = ((V_{e_9}, V_{e_2}), \text{ 绿色})$ 及 $e_D = ((V_{e_9}, V_{e_4}), \text{ 绿色})$ 来表示该组预留请求并添加至 G_D 上, G_D 更新为图11(d)。

[0209] 需要补充说明的是,本发明实施例中用来表示序号的参数下标i、k、M等均为自然

数。

[0210] S4,针对每个集群,在确定其内任一AGV依据对应的预留路径表行驶至与当前路径终点的距离达到预设距离值时,判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突。

[0211] 其中,预设距离值大于或等于AGV最短停车距离。

[0212] 可选的一种实施方式中,判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,包括:

[0213] 判断该AGV的下一条路径对应的边集合中的每一条边,是否在该集群内其余任一AGV的当前路径对应的边集合中;

[0214] 若是,判定该AGV的下一条路径与该集群内其余AGV的当前路径冲突,反之,判定该AGV的下一条路径与该集群内其余AGV的当前路径不冲突。

[0215] 如前,AGV的预留路径表中每一条路径可能包括道路网络中的一条或者多条边,还包括对应的冲突边,因此该AGV的下一条路径对应有一个边集合,同样的,该集群内其余每个AGV依据对应的预留路径表,当前路径也对应有一个边集合。如果,该AGV的下一条路径对应的边集合与该集群内其余任一AGV的当前路径对应的边集合中存在重复的边,则判定该AGV的下一条路径与该集群内其余AGV的当前路径冲突;反之,则判定该AGV的下一条路径与该集群内其余AGV的当前路径不冲突。

[0216] 若是,执行S5,向该AGV发出停止通行指令。

[0217] 具体的,该AGV收到停止通行指令后会停止在下一条路径的起点之前,即不驶入下一条路径,如此则可以避免与出现在下一条路径上的其余AGV发生冲突。

[0218] 若否,执行S6,向该AGV发出下一条路径允许通行指令。

[0219] 具体的,该AGV收到下一条路径允许通行指令后驶入下一条路径,说明该AGV不会在下一条路径上与其余AGV发生冲突,可以直接通行。

[0220] 因此,可以看出,每个AGV是利用实时更新的预留路径表,在受控情况下以多段分路径的形式行驶的。直观来看,AGV针对每一时刻的预留路径表在行驶时,对每一路并不直接通行,而是在接近路径终点时会接收中央控器系统发送的是否可以继续通行的相关指令,若AGV没有收到停止通行指令则会沿着预留路径表的下一个路径继续通行,若是AGV收到停止通行指令则会停止等待。因此,中央控器系统通过控制集群内各个AGV在路径上的停止与否,能够避免各个AGV在相关路径上发生冲突或者死锁。

[0221] 可选的一种实施方式中,向该AGV发出下一条路径停止通行指令之后,方法还包括:

[0222] 响应于该AGV的定时请求,再次判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,并根据判断结果向该AGV发出相应的指令。

[0223] 具体的,收到停止通行指令后,该AGV会停止在下一条路径的起点之前,为了继续完成运输任务,该AGV可以以一定的频率向中央控制系统发送请求,用于询问下一条路径是否可以通行。中央控器系统收到该AGV的请求后,需要按照S4中的判断方式,再次判断该AGV的下一条路径是否与该集群内其余AGV的当前路径冲突,如果冲突,则再次向该AGV发出停止通行指令;如果不冲突,则向该AGV发出下一条路径允许通行指令。

[0224] 为有效改善自动化码头大规模AGV集群运输场景下面临的道路拥堵,甚至车辆冲

突和死锁问题。本发明实施例提出了两阶段决策框架,设计了一种自动化码头的AGV在线路径规划方法,将原有路径规划问题拆解为在线线路规划与分布式轨迹规划两个紧密关联的子问题。在第一阶段,考虑水平运输路网中每段车道上的实时负载,以及相邻两边的转弯时间成本、平衡负载成本和预防冲突死锁成本,提出了改进型A^{*}算法在线计算AGV执行运输等任务从起点至终点的最优线路,通过平衡路网中的车流,减小变道转弯,降低道路拥堵以及死锁的风险;在第二阶段,提出了分布式轨迹规划方法,在位置发生变动时,基于AGV实时状态,通过计算各AGV在预设阈值内的预计通行区域,将多个AGV动态地划分为多个相互独立的集群,并利用分布式计算方式,在集群内独立计算各AGV的预留路径表以规划AGV预留路径,在集群内的AGV沿着预留路径表的轨迹行驶至接近当前路径终点时,通过判断AGV的下一条路径是否与集群内其余AGV的当前路径冲突给出是否可以继续通行的指令,因而可以保证集群内的AGV无冲突无死锁通行。并且,第二阶段通过将AGV按实时空间位置拆分成多个集群,降低了参与轨迹规划的AGV数量以及路线长度,降低了计算复杂度;而且,由于集群之间在空间上距离较远,各个集群内部的轨迹规划不会影响其他AGV行驶,通过集群管理实现了分布式改进,可以将互相独立的计算任务部署在不同的计算资源上进行运算,提高了计算效率。

[0225] 仿真实验结果表明,本发明实施例方法在有效改善道路拥堵情况,并保证行驶过程全程无冲突无死锁的情况下,能够求解规模更大的AGV集群的在线路径规划问题;同时,在线优化能够更好地应对AGV行驶过程中的不确定因素,采用分布式计算具有更高的运算效率,使AGV在复杂工业场景中的大规模应用成为可能。

[0226] 为了验证本发明实施例方案的效果,以下对仿真实验的结果进行分析说明。

[0227] (一) 实验设计

[0228] 自动化码头在完成建设、调试并投入使用后,将保持24小时连续运作,因而码头营业企业难以直接部署新算法或者新策略来测试其对实际生产的影响。为此,本发明实施例实验搭建了能够高度近似地模拟AGV在水平运输网络中行驶的仿真平台。该仿真平台能够模拟运输任务的产生与指派,以及AGV的控制(如加速、减速、匀速行驶、转弯等)。结合本发明实施例提出的决策框架,该仿真平台实现了对大规模AGV集群的模拟与优化调度,适用于验证两阶段框架与算法的可行性与性能。

[0229] 本发明实施例实验以真实码头布局为基础,生成了包含269个逻辑点、914条边及对应的冲突边的小规模场景用于算法测试。随后,以一个岸线长度为1公里,码头纵深约400米的真实码头为蓝图,基于实际施工方案建立了包含2250个逻辑点和10064条边的大规模电子地图,如图12所示,图12为本发明实施例实验中大规模电子地图的示意图。算法与仿真均在Visual Studio 2019环境下采用C#语言编程,所有的算例测试均在配置了8核2.30GHz的英特尔i7处理器及16GB内存的台式电脑上完成,操作系统为Windows10 64位。

[0230] 实验分为两个部分,首先验证改进型A^{*}算法对于线路选择结果以及车辆运行效率的影响。之后,分别验证分布式轨迹规划算法相较于集中式性能的提升,以及两阶段决策框架与算法在大规模场景下的有效性。为了测试算法性能,仿真模拟实际场景中运输任务在岸边或堆场出现的特点,所有的测试方案均通过在地图边界位置随机产生起点和终点的方式产生AGV的运输任务,产生时间服从一定的时间间隔,系统中所有AGV依次接受并执行运输任务。

[0231] (二) 改进型A^{*}算法的验证分析

[0232] 本实验首先在不考虑AGV行驶轨迹的情况下,探究改进型A^{*}算法对线路选择的影响;其次结合AGV仿真模拟平台,结合车辆行驶过程研究改进型A^{*}算法对车辆行驶效率的影响。

[0233] (1) 改进型A^{*}算法对线路选择的影响

[0234] 本实验首先在小规模场景下进行实验,分别在10辆、50辆、100辆、150辆及200辆AGV的实验规模下产生五组随机算例进行线路规划。为分别探究转弯时间系数B₁(B₁分别取0与0.3,B₂和B₃取0)、拥堵时间系数B₂(B₂分别取0与0.3,B₁和B₃取0)、冲突时间系数B₃(B₃分别取0与0.3,B₁和B₂取0.3)对于线路规划的影响。其中,单条边负载均值仅对有车辆(即AGV)使用的边进行平均,其数值等于路网中所有边上的负载之和除以车辆使用的边的数量。

[0235] 转弯时间系数B₁对线路规划的影响实验结果如图13所示。图13为本发明实施例实验中转弯时间因素对于单条线路直行边选择次数及转弯弧度均值的影响。当车辆数量相同时,改进型A^{*}算法选择直行边的次数较标准算法平均提高了2次以上,也就意味着在线路选择时,改进型A^{*}算法更倾向于优先选择直行边。同时,改进后的算法显著降低了转弯弧度均值,意味着车辆在线路选择时尽可能避免大角度变道与转弯。而当车辆数量增加时,以上指标并没有发生显著变化。

[0236] 负载及冲突边集负载的设置对线路规划的影响实验结果如图14和图15所示。通过图14可知,受拥堵时间系数B₂影响的改进型A^{*}算法能够显著减小单条线路负载均值与最大值,且随着车辆数量增加,改进效果越好。图15表明,受冲突时间系数B₃影响的改进型A^{*}算法能够在一定程度上降低与单条线路有关联的冲突边集的负载均值与最大负载,而改进效果随着车辆数量增加而略有提升。其中,200辆车的规模下,负载的设置对于单条线路平均负载的改善率达到44.52%,冲突边集负载的设置对于单条线路负载均值的改善率达到11.24%。以上结果意味着,改进型A^{*}算法在线路选择时将车流分散在更多备选线路上,降低在同一条边上或者冲突边上安排车辆的次数,有利于在后续车辆运行过程中减少拥堵概率和冲突概率,进而提升了轨迹规划算法的求解效率。

[0237] (2) 改进型A^{*}算法对车辆运行效率的影响

[0238] 结合AGV仿真模拟平台以及车辆动态行驶过程,本实验探究改进型A^{*}算法用于在线线路规划对于AGV运输效率的影响,并使用传统A^{*}算法进行对比,分析对于AGV平均行驶时间和任务全部完成时间的影响。其中改进型A^{*}算法中转弯时间系数B₁、拥堵时间系数B₂和冲突时间系数B₃的值均设置为0.3。实验在小规模场景下进行,算例规模分别设置为10辆、30辆、50辆、75辆、100辆AGV。实验设置与结果统计如表1所示:

[0239] 表1A^{*}算法与改进型A^{*}算法性能比较

项目	算法	10辆	30辆	50辆	75辆	100辆
[0240] AGV 平均行驶时间 (秒)	A^*	377.6	419.83	420.42	471.33	494.2
	改进型 A^*	374	411.767	413.28	442.389	446.77
	改善率	0.95%	1.92%	1.70%	6.14%	9.60%
全部任务完成时间 (秒)	A^*	683	1097	1599	2219	2598
	改进型 A^*	677	1081	1504	2053	2571
	改善率	0.88%	1.46%	5.94%	7.49%	1.04%

[0241] 通过实验结果可以看出,改进型A*算法通过降低变道转弯、使用动态负载分散车流、考虑冲突边集的负载以降低死锁发生概率,有利于缩短AGV单次平均运行时间以及任务总完成时间,提高了单个AGV以及AGV整体系统的运行效率。其中,随着AGV数量的增多,AGV平均行驶时间改善率有逐渐增加的趋势,在100辆车的实验规模下,改善率高达9.6%,说明在AGV数量规模较大时,车辆的转弯次数多,拥堵冲突现象更加严重,而减少转弯、分散车流、减少拥堵与冲突,对于提高AGV系统运行的效率更有帮助,也说明了本发明实施例改进A*算法目标优化的必要性。

[0242] (三) 分布式轨迹规划算法的验证分析

[0243] 为了探究算法的有效性,本实验首先基于小规模场景开展,对集中式与分布式的轨迹规划算法的性能进行对比,在10~100辆的AGV数量范围内,将算法检查状态更新的时间间隔 Δt 设置为1秒,以及预设时间阈值设置为140秒进行数值实验;之后在大规模场景下,测试AGV数量在100~500辆时算法的有效性。

[0244] (1) 集中式与分布式的轨迹规划算法性能对比

[0245] 首先,实验分别使用集中式算法与分布式算法求解AGV无死锁轨迹规划,分别在10辆、20辆、30辆、50辆、75辆、100辆AGV的算例规模下进行实验。其中,集中式决策方式在新的任务产生时,将系统中的所有AGV作为一个集群,对从起点至终点的线路全长计算无死锁的预计通行方案。

[0246] 实验选取了如表2所示的7个评价指标。其中,无死锁通行方案决策次数表示执行轨迹规划算法的次数,也表示集群变动或更新的次数;决策时间均值与最大值表示各集群执行轨迹规划算法的时长均值与最大值,其代表算法的时间效率;线路平均预留长度与最大长度表示所有车辆行驶全过程中的预留路径表分段的平均长度和最大长度,其数值越小表示车辆在行驶过程中等待的概率和时间就越短,实验结果统计如表2所示。

[0247] 表2集中式与分布式决策性能比较

项目	算法	10辆	20辆	30辆	50辆	75辆	100辆
[0248] AGV 平均行驶时间 (秒)	集中式	451.4	572.6	-	-	-	-
	分布式	361	430	426.3	425.7	421.48	450.7

[0249]	全部任务完成时间 (秒)	改善率	19.94%	24.90%	-	-	-	-
		集中式	814	1556	-	-	-	-
		分布式	680	977	1041	1507	2086	2496
		改善率	16.46%	37.21%	-	-	-	-
	无死锁通行方案决策次数(次)	集中式	10	20	-	-	-	-
		分布式	144	1039	1818	4364	6676	9092
	决策时间均值 (秒)	集中式	0.010548	0.011950	-	-	-	-
		分布式	0.001789	0.002842	0.002690	0.002268	0.002282	0.0019179
		改善率	83.04%	76.22%	-	-	-	-
	线路平均预留长度 (条边)	集中式	1	1.001125	-	-	-	-
		分布式	1	1	1.001850	1.002438	1.000062	1.005387
		改善值	0	0.001125	-	-	-	-
	预留区域最大长度 (条边)	集中式	1	3	-	-	-	-
		分布式	1	1	7	5	2	7
		改善值	0	2	-	-	-	-

[0250] 从实验结果可知,集中制轨迹规划算法在当前地图规模下只能处理20辆以下的AGV死锁预防问题,而分布式轨迹规划算法针对10~100辆AGV的实验规模下的死锁预防问题均可以正常计算,验证了本发明实施例设计的分布式轨迹规划算法的有效性。而且,根据实验结果可以得知,分布式轨迹规划算法有利于降低AGV平均运行时间,缩短任务全部完成时间,降低平均单次决策时间,降低线路平均预留长度与最大长度。在10辆和20辆AGV的实验规模下,AGV平均行驶时间平均改善率在20%左右;全部任务完成时间平均改善率分别为16.46%和37.21%;决策时间均值改善率为76.22%和83.04%。实验结果说明了集中式无死锁方案过于保守,导致较多的等待时间,分布式算法能够有效提高运输效率;对车辆进行集群管理并分散计算任务,能够加快决策效率。同时需要关注,分布式轨迹规划算法的决策次数较多,因此在实际应用过程中依赖于中央控制系统配备更多的计算资源来实现分布计算。

[0251] (2) 大规模场景下分布式轨迹规划算法的有效性验证

[0252] 随着自动化码头规模不断扩大,为验证本发明实施例设计的两阶段算法对于大规模AGV系统的实用性和有效性,本发明实施例模拟真实的码头水平布局,设计了2250个逻辑点和10064条连通边的水平运输路网,对100辆、200辆、300辆、500辆AGV规模的随机算例进行测试,实验结果如表3所示。

[0253] 表3大规模AGV运输系统测试结果

项目	100 辆	200 辆	300 辆	500 辆	
[0254]	AGV 平均行驶时间 (秒)	276.4	309.5	321.5	314.0
	全部任务完成时间 (秒)	2400	4901	6954	10580
	无死锁通行方案决策次数 (次)	6924	20056	28570	50308
	决策时间均值 (秒)	0.002108	0.001932	0.001883	0.002291

[0255]	决策时间最大值 (秒)	0.341438	0.183006	0.132071	0.513882
	线路平均预留长度 (条边)	1.000862	1.000764	1.001475	1.003382
	预留区域最大长度 (条边)	7	8	11	13

[0256] 实验结果显示,所有车辆均能达到任务的终点,说明AGV在行驶过程中能够保证无冲突与无死锁。结果表明,本发明实施例提出的决策框架及相应算法能够在大规模自动化码头场景中解决100~500辆AGV数量规模的线路规划与安全运输的问题,单次决策时间平均值均在0.001秒的级别,最大决策时间也在秒级以内,验证了本发明实施例所研究的决策框架及优化算法的有效性,也说明了本发明实施例算法应对大规模AGV系统的路径规划问题时,能够满足其实时决策的要求。需要说明的是,当地图的规模较大且任务的起点终点位置随机产生时,平均行驶时间、全部任务完成时间等指标将会收到较大影响。

[0257] 需要说明的是,图13~图15中,A*算法简写为A*,改进型A*算法简写为改进A*。

[0258] 综上,通过大量的数值实验可以得知,本发明实施例第一阶段的改进型A*算法能够实现减少变道转弯、分散路网车流、平衡冲突边负载的线路优化目标;第二阶段的分布式轨迹规划方法能够大幅度缩短车辆轨迹规划所需的决策时间,克服了传统轨迹规划算法无法应用于大规模AGV系统的问题。实验结果表明本发明实施例所提出的方法能够保障100~500辆AGV无死锁、安全且高效地运行,且算法平均计算时间均在0.001秒级别的水平,具有较高的计算效率。

[0259] 以上所述仅为本发明的较佳实施例而已,并非用于限定本发明的保护范围。凡在本发明的精神和原则之内所作的任何修改、等同替换、改进等,均包含在本发明的保护范围内。

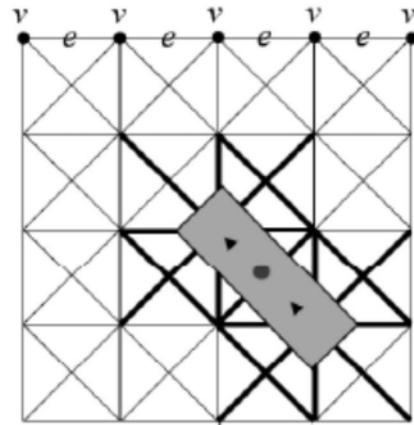


图1

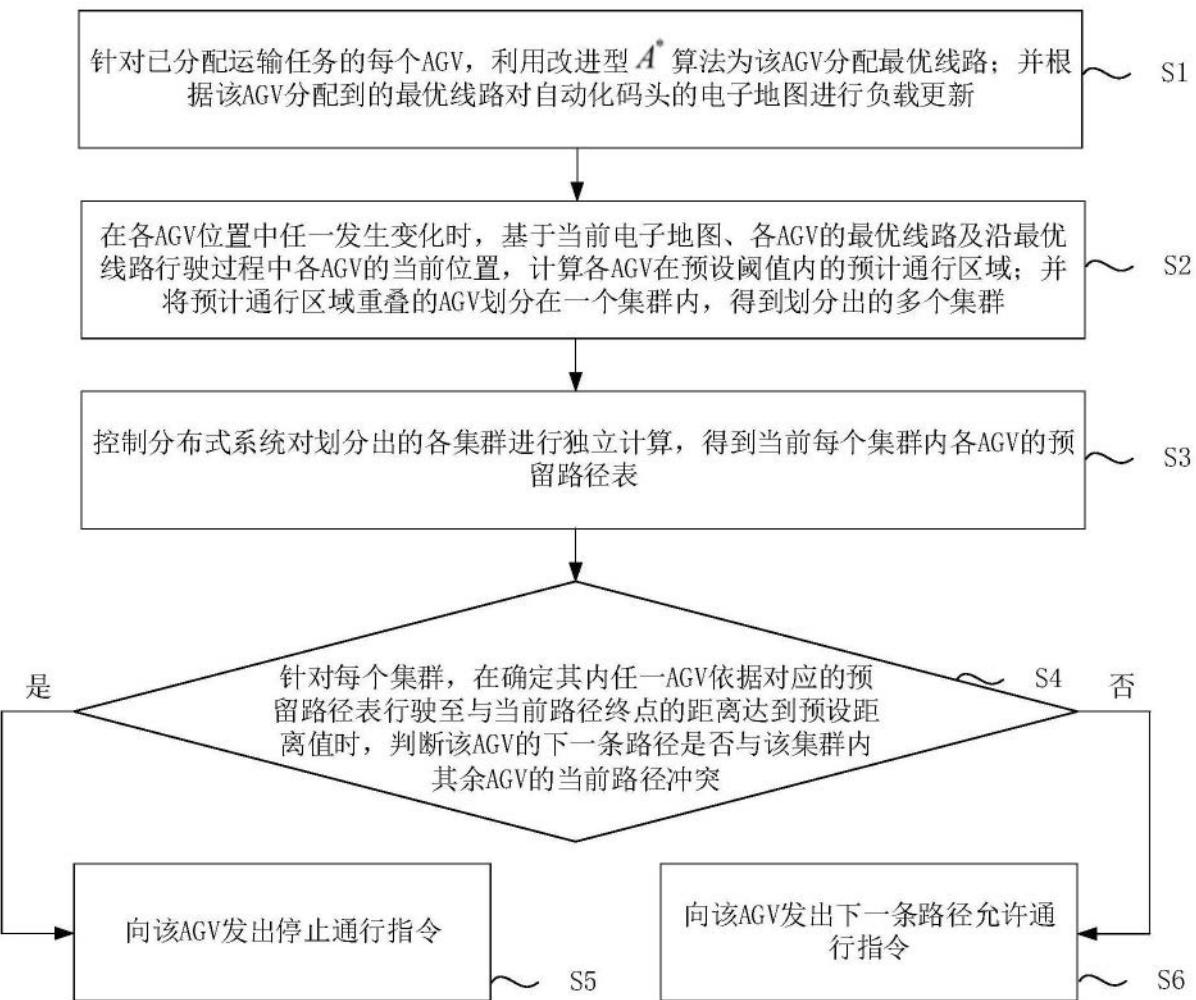


图2

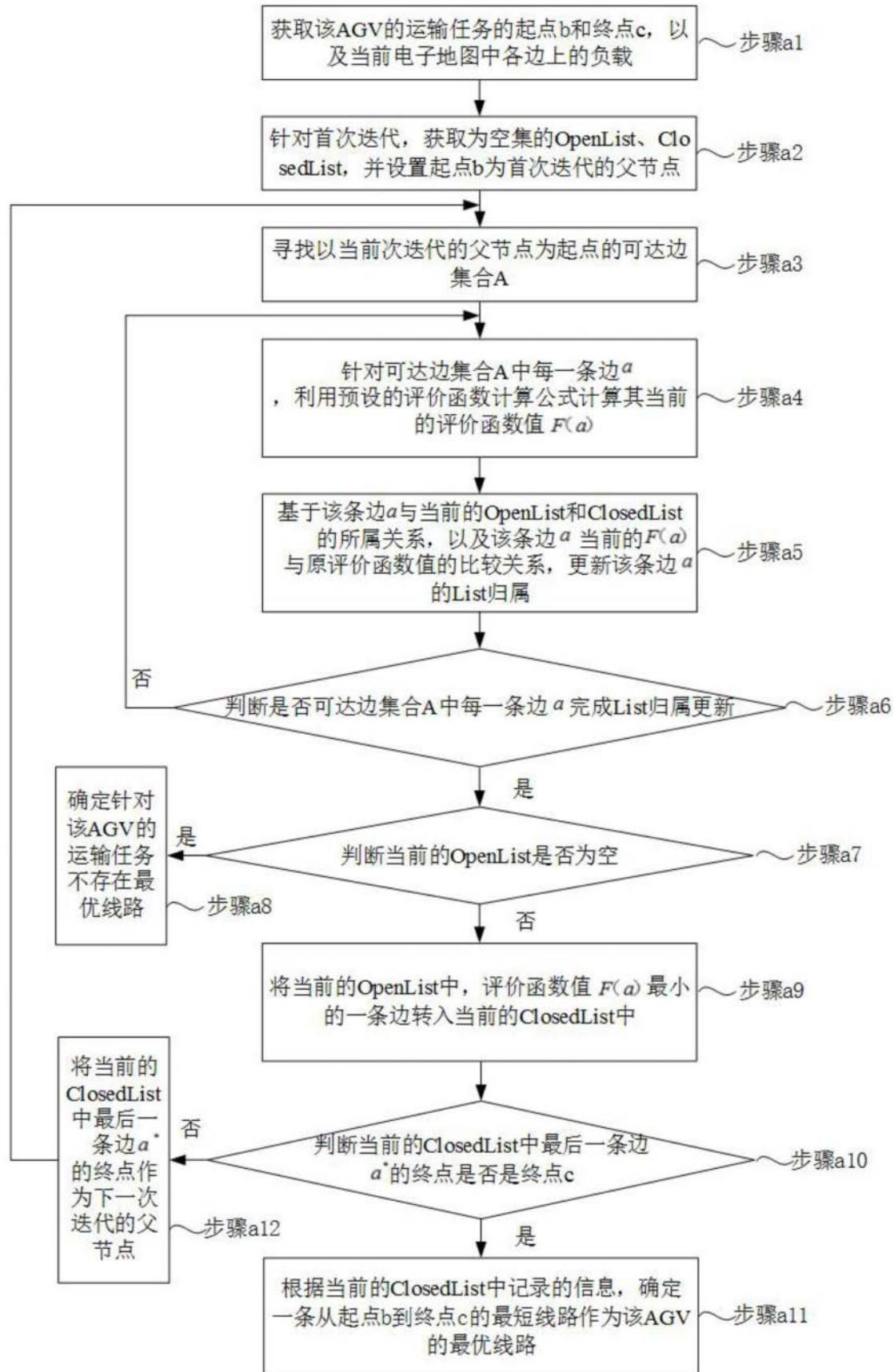


图3

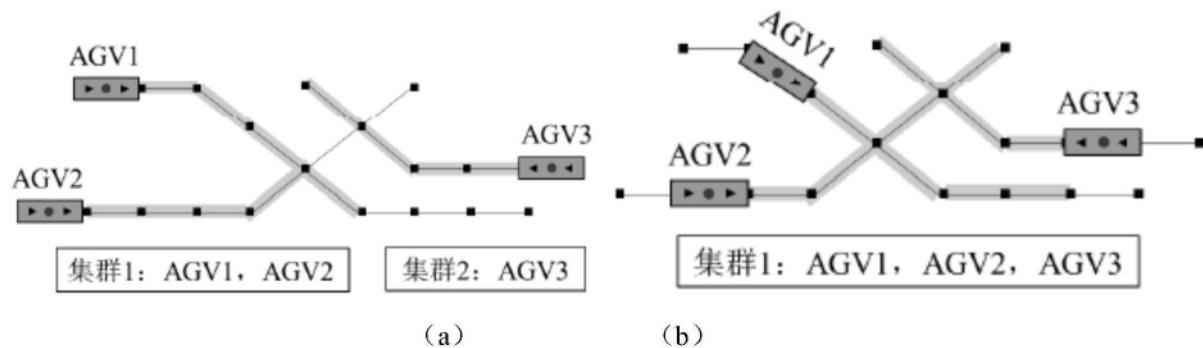


图4

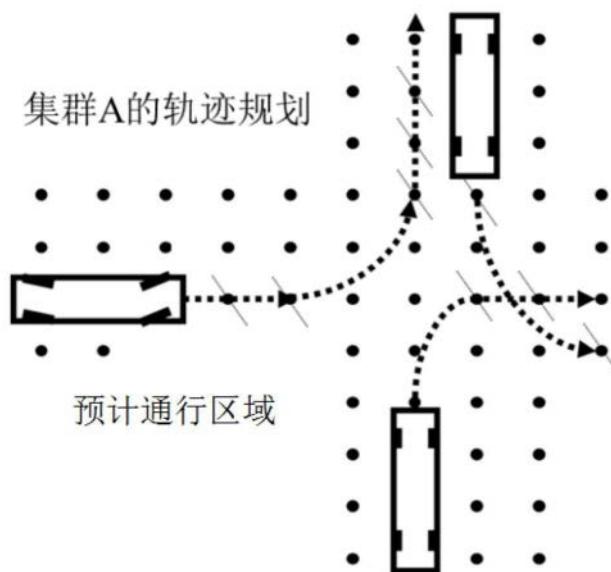


图5

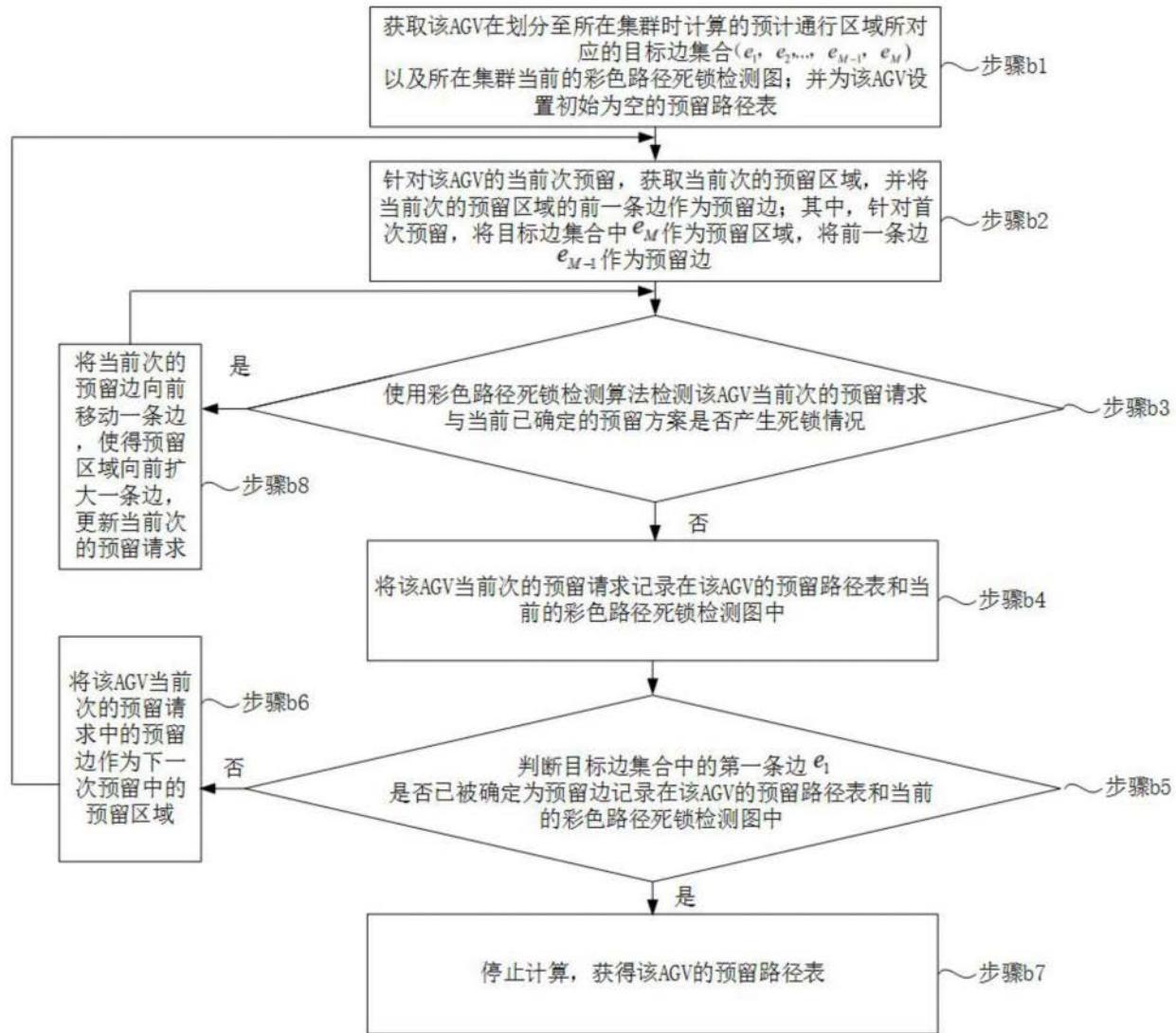


图6

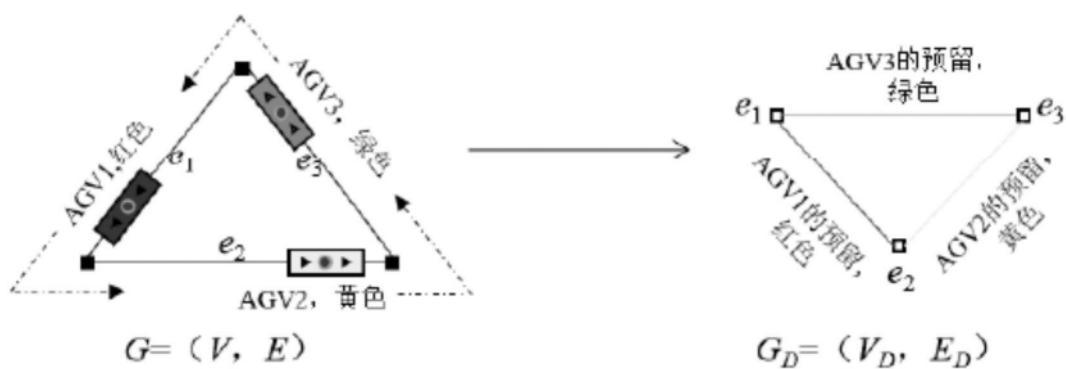


图7

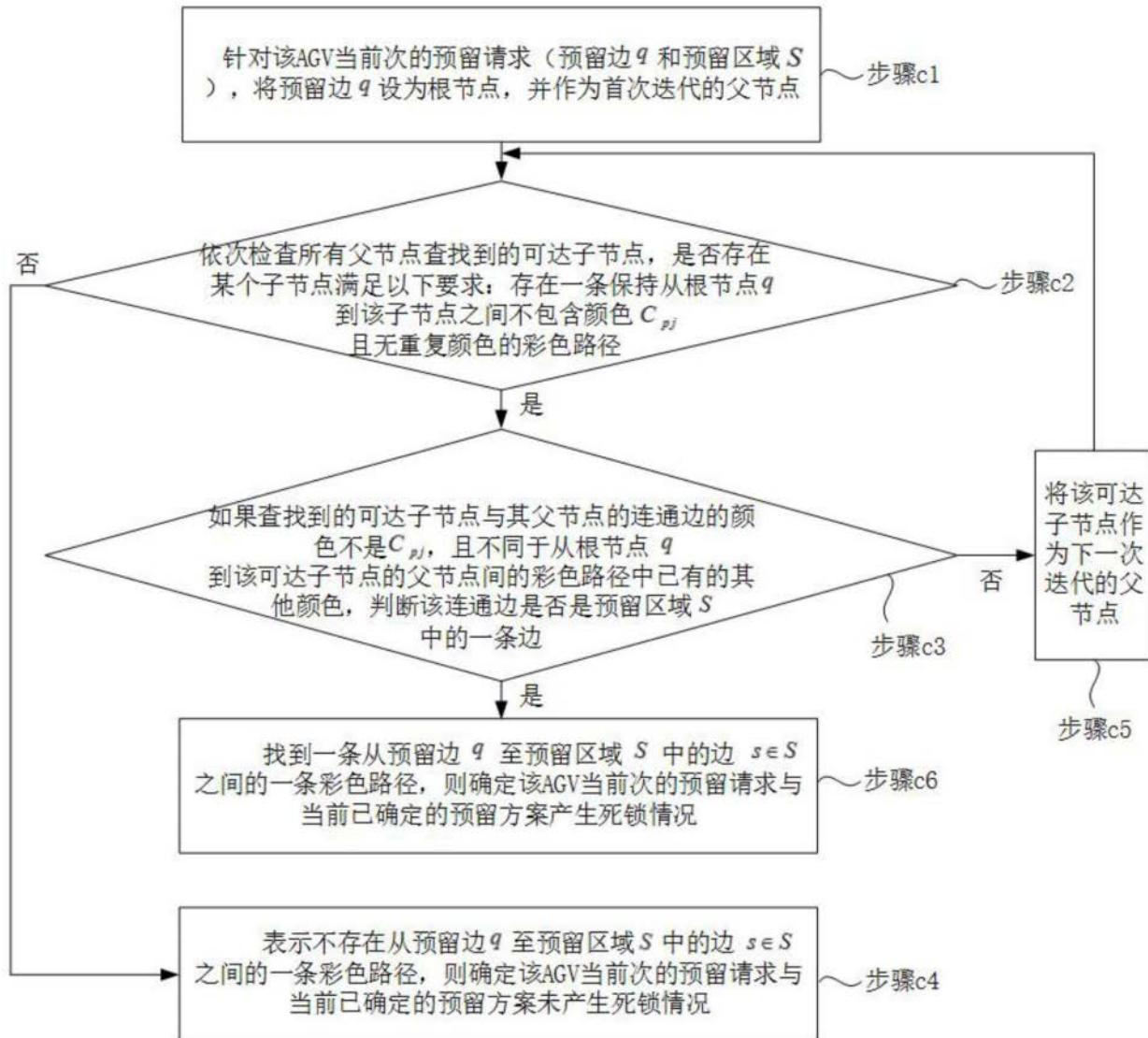


图8

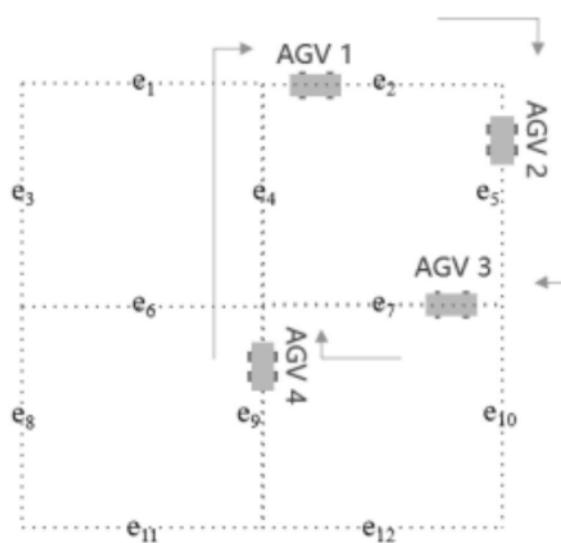


图9

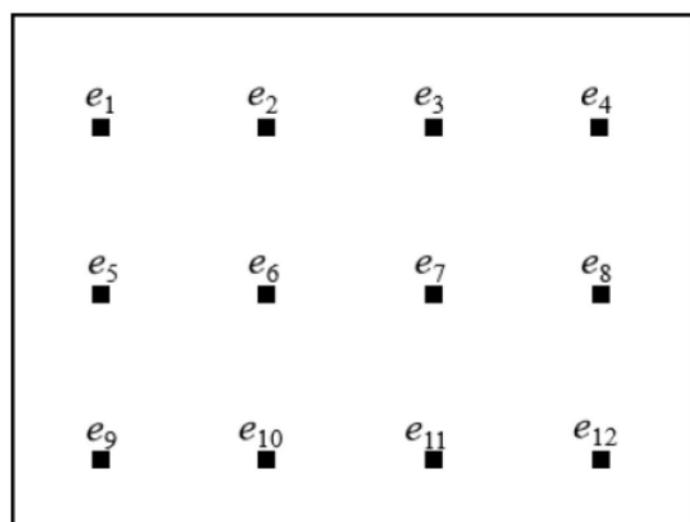


图10

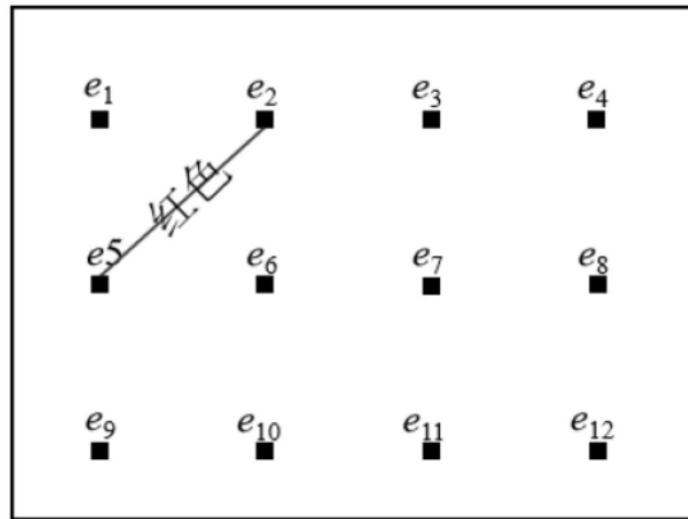


图11 (a)

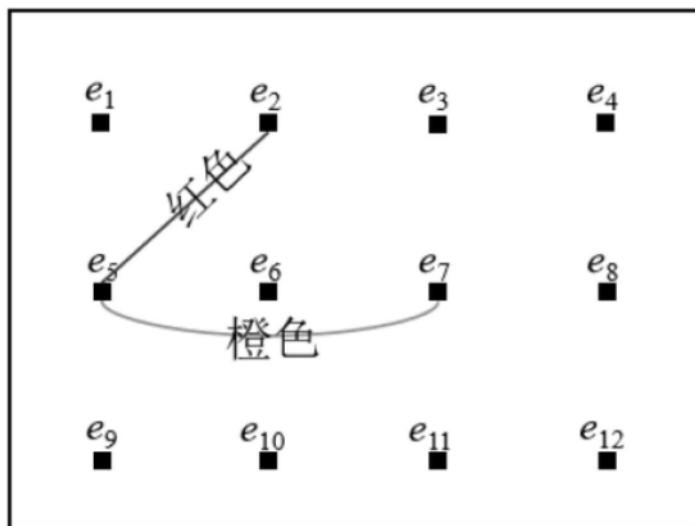


图11 (b)

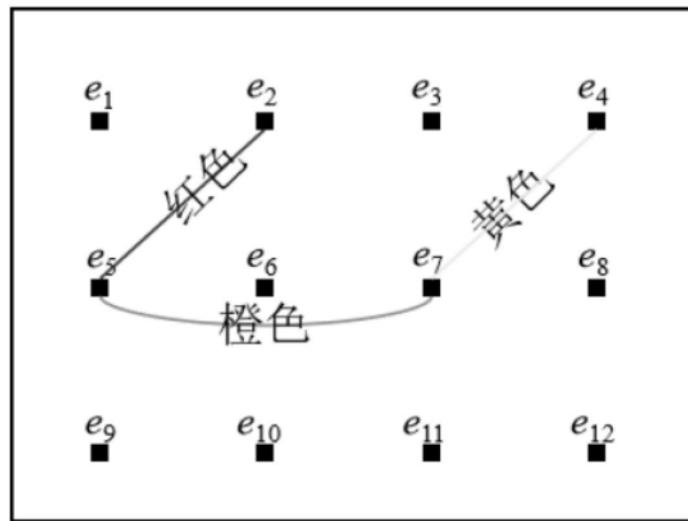


图11 (c)

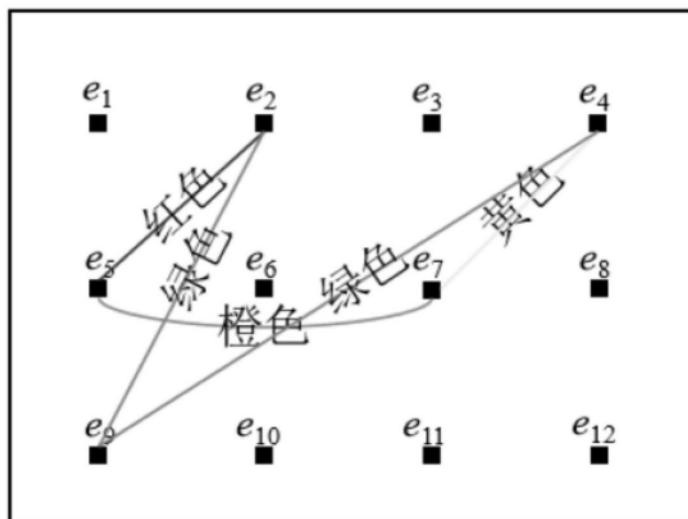


图11 (d)

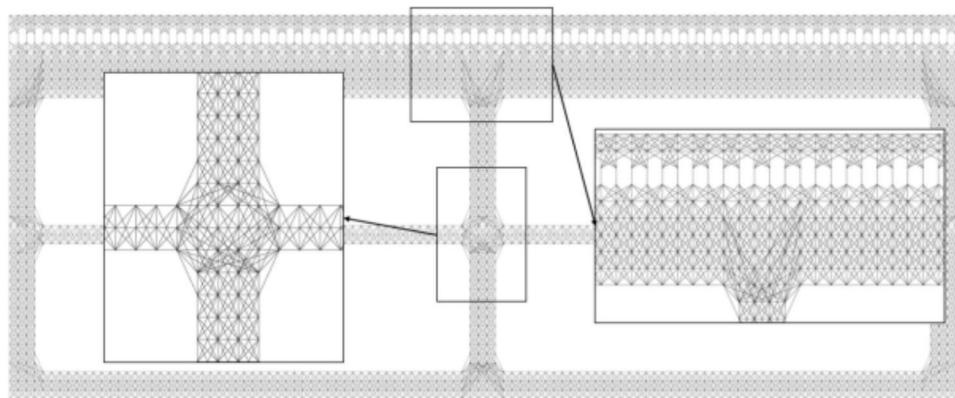


图12

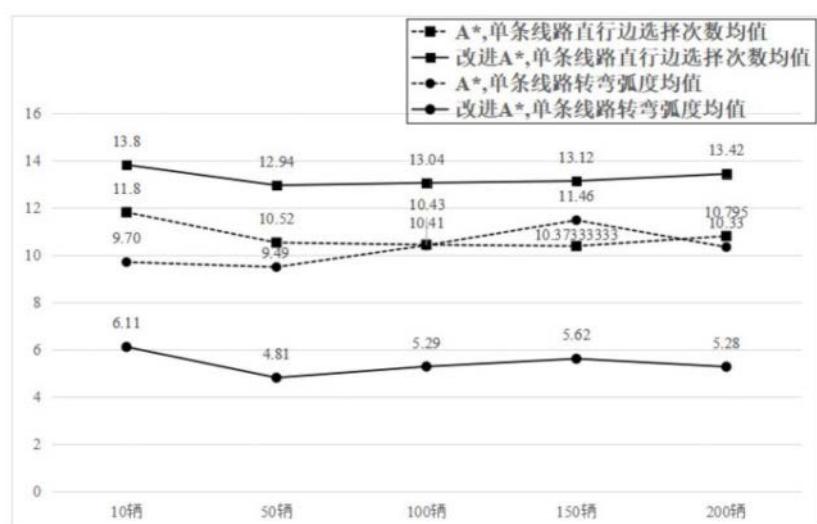


图13

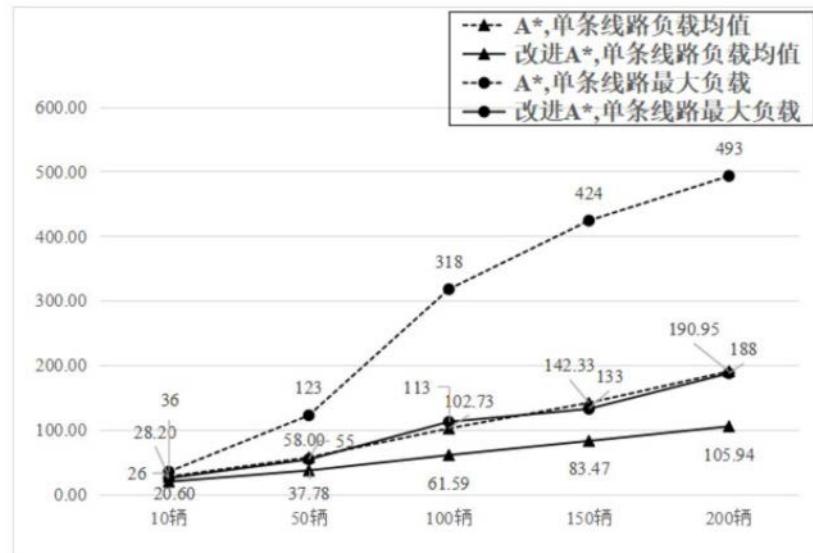


图14

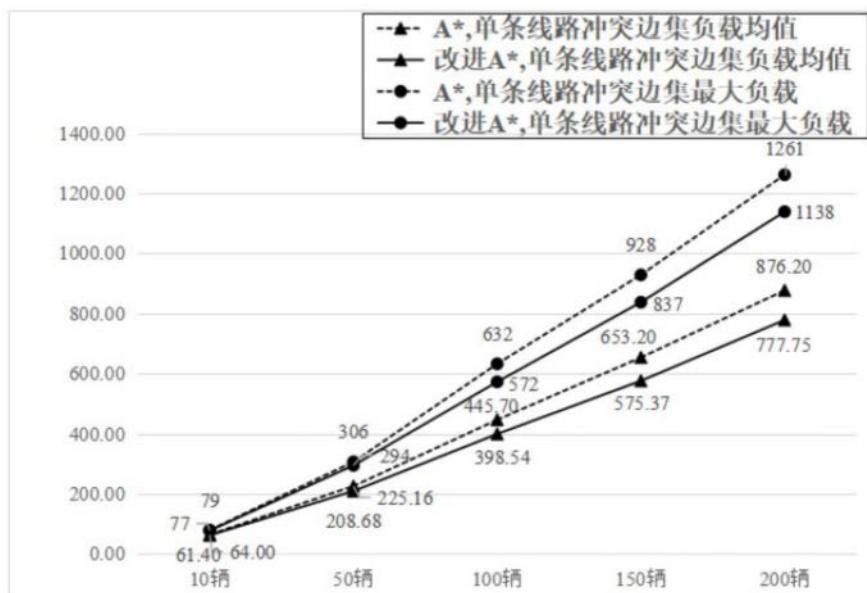


图15