

2-15-2024

## Simulation Platform of AGV System Scheduling Algorithms in Uncertain Environment

Zhihao Shi

*Sino-US Global Logistics Institute, Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China, shizh825@sjtu.edu.cn*

Haihui Shen

*Sino-US Global Logistics Institute, Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China; Data-Driven Management Decision Making Lab, Shanghai Jiao Tong University, Shanghai 200030, China, shenhaihui@sjtu.edu.cn*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact [xtfzxb@126.com](mailto:xtfzxb@126.com).

---

## Simulation Platform of AGV System Scheduling Algorithms in Uncertain Environment

### Abstract

**Abstract:** As a complete automated guided vehicle(AGV) system scheduling algorithm, it must include conflict-handling strategy, together with common dispatching strategy and routing algorithm. However, due to the uncertainty, characteristics of such scheduling algorithm are difficult to be analyzed theoretically, and the relevant study is lacking. An AGV system simulation platform based on the discreteevent simulation technique is designed and developed, which can flexibly set the scheduling problem, choose the dispatching strategy, routing algorithm, and conflict-handling strategy for the scheduling algorithm, to run the simulation. The platform has a visual interface, from which the running status of AGVs and the performance of the scheduling algorithm can be intuitively observed, and it can also output statistical data of the simulation experiment. Experiment results show that the simulation platform can accurately and effectively evaluate and compare the scheduling algorithms in uncertain environment.

### Keywords

automated guided vehicle(AGV) system scheduling, uncertain environment, route conflict, discrete-event simulation, simulation platform

### Recommended Citation

Shi Zhihao, Shen Haihui. Simulation Platform of AGV System Scheduling Algorithms in Uncertain Environment[J]. Journal of System Simulation, 2024, 36(2): 385-404.

## 不确定情景下 AGV 系统调度算法的仿真平台

石志浩<sup>1</sup>, 沈海辉<sup>1,2\*</sup>(1. 上海交通大学 安泰经济与管理学院 中美物流研究院, 上海 200030;  
2. 上海交通大学 “数字化管理决策” 教育部哲学社会科学实验室, 上海 200030)

**摘要:** 不确定情境下的完整自动引导车(automated guided vehicle, AGV)系统调度算法, 除了要包含常见的任务指派策略和路径规划算法之外, 还需要包含冲突应对策略。由于存在随机性, 这样的调度算法难以从理论上分析其特性, 相关研究也较少。基于离散事件仿真技术, 设计并开发了一个 AGV 系统仿真平台, 可以灵活地设置调度问题, 并选择调度算法中的任务指派策略、路径规划算法和路径冲突应对策略进行仿真。该平台具有可视化的界面, 可以直观地观察 AGV 的运行状态和调度算法的表现, 也可以输出最终的仿真实验统计数据。实验表明: 该平台可以准确有效地对不确定情景下的调度算法进行评估与比较。

**关键词:** AGV(automated guided vehicle)系统调度; 不确定情景; 路径冲突; 离散事件仿真; 仿真平台  
中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X(2024)02-0385-20

DOI: 10.16182/j.issn1004731x.joss.22-1214

**引用格式:** 石志浩, 沈海辉. 不确定情景下 AGV 系统调度算法的仿真平台[J]. 系统仿真学报, 2024, 36(2): 385-404.

**Reference format:** Shi Zhihao, Shen Haihui. Simulation Platform of AGV System Scheduling Algorithms in Uncertain Environment[J]. Journal of System Simulation, 2024, 36(2): 385-404.

## Simulation Platform of AGV System Scheduling Algorithms in Uncertain Environment

Shi Zhihao<sup>1</sup>, Shen Haihui<sup>1,2\*</sup>(1. Sino-US Global Logistics Institute, Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China;  
2. Data-Driven Management Decision Making Lab, Shanghai Jiao Tong University, Shanghai 200030, China)

**Abstract:** As a complete automated guided vehicle(AGV) system scheduling algorithm, it must include conflict-handling strategy, together with common dispatching strategy and routing algorithm. However, due to the uncertainty, characteristics of such scheduling algorithm are difficult to be analyzed theoretically, and the relevant study is lacking. An AGV system simulation platform based on the discrete-event simulation technique is designed and developed, which can flexibly set the scheduling problem, choose the dispatching strategy, routing algorithm, and conflict-handling strategy for the scheduling algorithm, to run the simulation. The platform has a visual interface, from which the running status of AGVs and the performance of the scheduling algorithm can be intuitively observed, and it can also output statistical data of the simulation experiment. Experiment results show that the simulation platform can accurately and effectively evaluate and compare the scheduling algorithms in uncertain environment.

**Keywords:** automated guided vehicle(AGV) system scheduling; uncertain environment; route conflict; discrete-event simulation; simulation platform

收稿日期: 2022-10-13 修回日期: 2022-12-02

基金项目: 国家自然科学基金(72001140, 72031006); 科技创新 2030 “新一代人工智能” 重大项目(2018AAA0101700)

第一作者: 石志浩(1997-), 男, 回族, 硕士, 研究方向为 AGV 系统的数字孪生与仿真。E-mail: shizh825@sjtu.edu.cn

通讯作者: 沈海辉(1989-), 男, 助理研究员, 硕士生导师, 博士, 研究方向为仿真建模、分析与优化, 及其在生产、管理、医疗等领域的应用。E-mail: shenhaihui@sjtu.edu.cn

## 0 引言

自动引导车(automated guided vehicle, AGV)是一种具有自动导引装置,能够沿设定的路径行驶且具有各种物料移栽功能的搬运车辆<sup>[1]</sup>。AGV 由美国伊利诺伊州诺斯布鲁克的巴雷特电子公司设计制造<sup>[2]</sup>。发展至今,AGV 被广泛用于生产物流与仓储服务领域<sup>[3-4]</sup>。2021 年,全球 AGV 市场价值约为 20 亿美元。根据 Fortune Business Insights 的预测,2022—2029 年间,全球 AGV 市场规模将以 9.6% 的年复合增长率从 21.7 亿美元增长到 41.1 亿美元<sup>[5]</sup>。此外,AGV 系统由于其运行状态和控制指令相对简单,较易实现数字孪生,许多企业都建立了不同程度的 AGV 系统数字孪生平台<sup>[6-9]</sup>。一方面,AGV 物理实体的运行信息(如位置、状态、任务等)可以实时同步更新到 AGV 数字孪生体上;另一方面,在数字空间中做出的最优决策可以即时转化为对 AGV 物理实体的控制指令。当然,如何利用好 AGV 数字孪生实现更加智能、有效与实用的决策,仍是需要持续探索的问题。

AGV 系统运行效率的核心是 AGV 调度算法,主要包括任务指派策略和路径规划算法。前者确定每个运输任务由哪辆 AGV 执行;后者决定 AGV 完成运输任务的行驶路线。由于 AGV 工作场景的特点,通常需要考虑多辆 AGV 在路径上的冲突问题。现有研究大多是在确定性的设定下,规划出没有路径冲突的最优路径。然而,在实际应用环境中,由于 AGV 的机械与控制的误差、场地误差、现场干扰等因素的存在,AGV 的实际行驶、作业时间往往会偏离理论值,从而导致 AGV 在按照规划路径行驶的过程中出现路径冲突。因此,对于一个实际的 AGV 系统来说,有效的调度算法不仅需要包含任务指派策略和路径规划算法,还需要包含冲突应对策略。对所有路径冲突都使用人工介入的处理方式,虽然简单但是低效。一种比较简单但相对智能的策略是对于发生路径冲突的 AGV 即时执行路径重规划。Mäntysaari<sup>[10]</sup>考

虑了在路径时间具有不确定的情况下,通过主动的周期性重规划来避免实际行驶时间与理论行驶时间之间偏差的积累,从而减缓路径冲突的发生。

目前,文献中针对不确定情景下 AGV 系统的调度算法(包含任务指派、路径规划和冲突应对)的研究还相对较少,对于上述提及的 2 种冲突应对策略或者它们的混合策略也缺乏评估和比较。这个问题的难点在于,在不确定情景下路径冲突的发生具有随机性,难以通过理论分析和计算刻画出某种冲突应对策略的效果与特性,只能诉诸于物理实验或者仿真实验。然而,当前主流的仿真软件<sup>[11]</sup>均无法直接实现对包含路径冲突应对策略在内的完整 AGV 调度算法进行建模与仿真。熊俊涛等<sup>[12]</sup>设计了一个基于交互式控制的 AGV 实时仿真系统,但关注点在于轨迹模拟与纠偏。不确定情景下 AGV 仿真的关键点和难点在于如何在程序中对路径冲突进行识别。在实际物理系统中,AGV 路径冲突由传感器检测,并发送中断信号至相关 AGV 的控制器。由于离散事件仿真技术<sup>[13]</sup>只在离散时间点上对系统状态进行更新,如何正确地使时钟推进至路径冲突发生的前夕并且判断路径冲突的类型,是一个值得研究的问题。本文通过将路径冲突归类和将 AGV 活动细分,确保对路径冲突的正确判断以及保证系统状态的唯一性。本文设计并开发了一个 AGV 系统仿真平台,该仿真平台可以设定路径网络、运输任务、AGV 数量、路径时间不确定性,以及随机障碍等,并选择调度算法中的任务指派策略、路径规划算法和路径冲突应对策略,然后运行仿真。该平台具有可视化的界面,可以直观地观察 AGV 的运行状态和整个调度算法的表现,也可以输出最终的仿真实验统计数据。

## 1 AGV 调度算法研究现状

### 1.1 任务指派研究

AGV 任务指派策略主要可分为任务驱动与车

辆驱动<sup>[14]</sup>。前者是指将新出现的运输任务分配给空闲AGV中的某一台, 适用于AGV系统负载率较低的场景; 后者是指将执行完任务的AGV分配给正在等待的运输任务中的某一个, 适用于AGV系统负载率较高的场景。如果在AGV系统开始运行前已知全部的运输任务, 那么运输任务集合与AGV集合就组成了一个完全二部图, 任务指派问题就可以转化为完全二部图的匹配问题, 最优解可以通过匈牙利算法求出<sup>[15]</sup>。多数情况下, 运输任务是在AGV系统运行过程中实时到来的, 此时的任务指派是一个在线实时问题。对于此类问题的研究主要集中在启发式规则的设计上。Bartholdi等<sup>[16]</sup>提出了先遇先服务策略, 并将其应用于单环线分散控制AGV运输系统。在此规则下, 运输任务会被优先指派给第一辆来到运输任务起点且有足够载货空间的AGV。仿真实验表明: 对于特殊AGV运输系统, 先遇先服务策略比其他启发式策略(如先到先服务策略)表现更好。Egbelu等<sup>[17]</sup>考虑一种工作中心控制AGV系统, 并从任务指派的2种类型出发, 基于距离、AGV使用率、运输任务等待时间等指标, 提出了包含最近车辆优先(nearest-vehicle-first, NVF)、最长空闲车辆优先(longest-idle-vehicle-first, LIVF)等在内的5种任务驱动的任务指派策略, 以及包含最近任务优先(nearest-mission-first, NMF)、最早任务优先(earliest-mission-first, EMF)等在内的7种车辆驱动的任务指派策略。仿真实验表明: 当AGV系统布局不满足某些特定的条件时, 基于距离的任务指派策略会有缺陷。文献[14]使用配送中心、生产工厂与集装箱中转码头3个真实案例的数据, 对经典的任务指派策略进行了仿真对比。实验结果表明: NVF与NMF等基于距离指标的任务指派策略的性能显著优于其他策略。

## 1.2 路径规划研究

AGV路径规划问题是VRP(vehicle routing problem)问题的一个变体, 但由于车辆及路径网络

在物理尺寸上的差异, VRP问题一般不考虑车辆在路径上的冲突问题, 而AGV路径规划问题则通常需要考虑路径冲突问题。Qiu等<sup>[18]</sup>将AGV路径冲突分为碰撞、拥挤、活锁、死锁4种。Möhring等<sup>[19]</sup>将AGV路径冲突分为死锁与活锁2种。现有AGV路径规划的研究中, 大多基于确定性路径时间假设, 来规划无冲突的路程最短路径。Broadbent<sup>[20]</sup>提出了AGV无冲突最短时间路径规划的概念。他们使用Dijkstra最短路径算法<sup>[21]</sup>生成一个描述车辆路径占用时间的矩阵, 通过比较路径占用时间来检测车辆之间的潜在冲突。Huang等<sup>[22]</sup>提出了双向路径网络中AGV最短时间路径的标记算法。该算法将原始路径网络的路径全部转换为节点, 然后为每个节点分配一个包含其空闲时间窗口的标签, 通过比较每个节点的标签, 可以得到一个无冲突最短时间路径。Mäntysaari<sup>[10]</sup>对Huang等的算法做了进一步研究, 借鉴了A\*算法<sup>[23]</sup>的思想来选择路径拓展基节点, 还在路径规划过程中对地图进行修剪, 进一步提高了算法效率。Taghaboni等<sup>[24]</sup>提出了单双向路径都可以使用的AGV路径规划动态算法。但此算法在任务和车辆数量增加时效率快速降低, 无法保证求得路径的最优性。如前文所述, 在实际应用环境中, 路径时间有时 would 存在由各种因素引发的不确定性。Mäntysaari<sup>[10]</sup>初步考虑了在路径时间具有不确定性的情况下, 通过周期性时钟同步与路径重规划来避免实际行驶时间与理论行驶时间之间偏差的积累, 以减缓路径冲突的发生。但这种方法不能完全避免路径冲突, 也无法对路径冲突进行实时处理。

## 1.3 路径冲突应对

现有针对AGV路径冲突的研究, 主要是从预防路径冲突发生的角度出发。1.2节中的确定性假设下的无冲突路径规划, 在不确定情景下无法避免冲突, Mäntysaari提出的周期性重规划策略也只能做到减少冲突发生。也有从路径网络设计与管

控层面来避免冲突的研究。例如, Sun等<sup>[25]</sup>使用基于深度优先搜索的分支定界算法, 考虑空车流量的影响, 规划设计了单向AGV路径网络。Malmberg<sup>[26]</sup>考虑了将AGV路径网络分成许多控制区, 一个控制区内同时只能容纳一辆AGV。这些方法需要AGV路径网络具有一些特殊的结构, 通用性差; 额外加入了AGV通行管控, 降低了路径网络利用效率。在不确定情景下, 更加通用且高效的AGV路径冲突应对策略, 应当是先在确定性假设下为AGV规划无冲突的最优路径, 甚至是忽略冲突的最优路径, 然后在不可避免的路径冲突发生之后, 实时地进行冲突解除, 也即动态调度的思想<sup>[27]</sup>。但是由于冲突应对策略(连同相应的任务指派策略与路径规划算法)的效果与特性难以通过理论分析和计算刻画, 目前文献中相关的研究还比较少。导致在实际应用中, 普遍采用人工介入处理的方式来应对路径冲突, 制约了AGV系统的自动化与智能化水平。

## 2 问题与模型设定

本文主要研究双向单通道布局的AGV系统, 因为它在制造生产、拣选配送与仓储物流等领域的应用更为广泛<sup>[28-30]</sup>。在该AGV系统中, 两节点间最多只有一条通道连接, AGV可以在每条通道上沿着两个方向中的任一方向行驶, 在任一节点上进行装/卸货操作。运输任务实时到来, 被指派的AGV首先从当前位置出发, 到运输任务起点装载货物, 然后将货物运送到终点, 最后卸载货物, 完成运输任务。一辆AGV一次最多只能执行一个运输任务。由于AGV的机械与控制的误差、场地误差、现场干扰等因素的存在, AGV的行驶速度/时间存在不确定性。此外, AGV路径网络中还可能实时地出现一些随机的障碍, 例如, 货物掉落或临时封闭部分区域等, 导致某些通道和节点在一段时间内不可用。在实际的AGV系统中, AGV的定位大多通过地面上位于节点处的二维码或RFID(radio frequency identification)实现, 因此,

控制系统无法精确定位到AGV在通道上的绝对位置。本文考虑的AGV系统也是如此, 对于AGV的定位仅限于在任意时刻AGV处于哪条通道或哪个节点, 而没有其在该通道或节点上的具体位置。

### 2.1 符号与模型

AGV路径网络可以被描述为网络 $G(N, A)$ , 其中,  $N$ 表示节点的集合,  $A$ 表示连接各节点之间弧(通道)的集合。AGV在整个网络中的标称行驶速度为 $v$ 。对于任意弧 $a \in A$ , 其长度为 $l_a$ , 可以各不相同。由于实际的AGV具有一定的物理尺寸, 它不仅需要时间通过一条弧, 也需要时间通过一个节点。为简便起见, 与文献[22]一样, 假定AGV沿弧 $a$ 从节点 $n_1$ 行驶至节点 $n_2$ 的过程中, 离开节点 $n_1$ 和进入节点 $n_2$ 均需要花费期望值为 $\delta$ 的时间, 而行驶在弧 $a$ 上的时间的期望为 $l_a/v - 2\delta$ 。换言之, 对于任意节点 $n \in N$ , AGV完整地通过它(先进入后离开)的时间期望为 $2\delta$ 。此外, 节点与弧的容量均为1。在不确定情景下, AGV正常通过节点 $n \in N$ 的时间为 $2\delta + \varepsilon_n$ , 正常通过弧 $a \in A$ 的时间为 $l_a/v - 2\delta + \zeta_a$ , 其中,  $\varepsilon_n$ 和 $\zeta_a$ 为随机偏差。在此设定下, 路径冲突既可以发生在弧上, 也可以发生在节点上。AGV的装货操作时间为 $\delta_1$ , 卸货操作时间为 $\delta_u$ 。 $V$ 表示系统中AGV的集合,  $V_{\text{busy}}$ 表示正在工作的AGV集合,  $V_{\text{idle}}$ 表示空闲的AGV集合。一个运输任务可表示为 $\{o, d, t\}$ , 其中,  $o$ 为运输任务起点,  $d$ 为运输任务终点,  $t$ 为运输任务到来时刻。

由于弧和节点都有通过时间并且容量都为1, 本文采用文献[22]的方法将原AGV路径网络 $G(N, A)$ 转换为等价的只有节点时间的网络 $G(N', A')$ , 将原网络中的弧 $a \in A$ 等效为一个新的节点 $n_a$ , AGV通过节点 $n_a$ 的时间为 $l_a/v - 2\delta + \zeta$ ; 新增2条弧, 将这个新节点与原弧 $a$ 的2个端点连接起来, AGV在新增的2条弧上的通过时间为0。在转换后的网络 $G(N', A')$ 上, AGV的通过时间只发生在节点(原节点或新节点)上, 在所有弧上的通

过时间都为0。如无特别说明,下文的讨论均对于等价的只有节点时间的网络 $G(N',A')$ 进行: $n \in N'$ 表示任意一个原节点, $n_a \in N'$ 表示任意一个新节点(由原弧 $a \in A$ 转化而来), $n' \in N'$ 表示任意一个一般化的节点。

### 2.2 任务指派与路径规划

本文考虑2种任务驱动的任务指派策略:NVF策略和LIVF策略,2种车辆驱动的任务指派策略:NMF策略和EMF策略<sup>[17]</sup>。一个完整的任务指派策略可由任意一个任务驱动指派策略加上任意一个车辆驱动指派策略组成。具体地说,当运输任务到来时,如果有AGV处于空闲,则使用NVF策略(将任务指派给 $V_{idle}$ 中离orig最近的AGV)或LIVF策略(将任务指派给 $V_{idle}$ 中等待时间最长的AGV);如果没有空闲AGV,则将运输任务缓存至任务池并当有AGV变为空闲时使用NMF策略(将所有等待任务中起点离该AGV最近的任务指派给该AGV)或EMF策略(将所有等待任务中到来时间最早的任务指派给该AGV)。

本文考虑2种确定性路径时间假设下的路径规划算法:时间窗无冲突最短路算法<sup>[10,22]</sup>和A\*算法<sup>[23]</sup>。在进行路径规划时,先忽略AGV通过节点所需时间的随机偏差,通过节点 $n \in N'$ 的时间为 $2\delta$ ,通过节点 $n_a \in N'$ 的时间为 $l_a/v - 2\delta$ ,然后使用上述2个算法中的任意一个规划出路径并下发给AGV。本文使用的时间窗无冲突最短路算法是对Huang等<sup>[22]</sup>提出的算法的一个改进,借鉴了A\*算法中的思想来选择路径拓展基节点。关于本文使用的时间窗无冲突最短路算法的详细描述与说明,请见附录1。为表述方便,本文使用的时间窗无冲突最短路算法为TWA(time-window conflict-free shortest route algorithm),A\*算法为ASA(A-star algorithm)。

### 2.3 冲突应对策略

#### 2.3.1 冲突分类与识别

在 $G(N',A')$ 中,由于每个节点的容量都为1,

因此,在AGV行驶过程中当某个节点上的AGV数量由1即将变为2的瞬间,判定AGV路径冲突发生。当 $AGV1 \in V_{busy}$ 完成其当前节点的行驶,即将进入规划路径的下一个节点时,若发现该节点正被占用,则路径冲突便发生了,共有5种情形,如图1所示。①障碍冲突:该节点被AGV以外的障碍物占用;②空闲AGV冲突:该节点被 $AGV2 \in V_{idle}$ 占用;③工作AGV活锁:该节点被 $AGV2 \in V_{busy}$ 占用,但AGV1可以等AGV2离开后再进入该节点;④工作AGV对头死锁:该节点被 $AGV2 \in V_{busy}$ 占用,且AGV1与AGV2相向而行;⑤工作AGV循环死锁:该节点被 $AGV2 \in V_{busy}$ 占用,AGV2的下一个节点被 $AGV3 \in V_{busy}$ 占用,……,由此构成了一个循环。

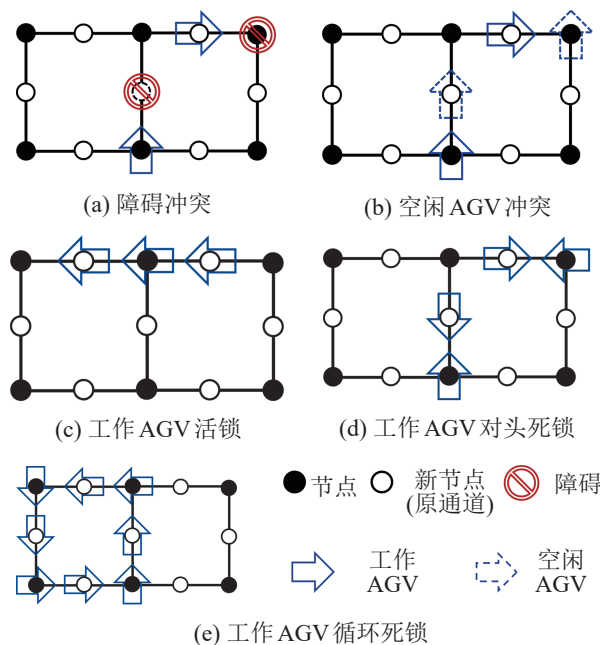


图1 路径冲突类型

Fig. 1 Types of route conflict

在程序中,障碍冲突与空闲AGV冲突只需要基于当前占用发生冲突节点的物体类别就可以识别。在由工作AGV造成的冲突中,如果要同时占用同一个节点的两AGV的行驶方向相反,则为工作AGV对头死锁;否则便为工作AGV活锁或工作AGV循环死锁。工作AGV活锁可以通过让后

方的 AGV 等待相继完成通行，而工作 AGV 循环死锁则会在等待的过程中出现死循环。

### 2.3.2 冲突应对

本文首先考虑一种比较简单的策略，称之为冲突解决型策略(conflict-solving strategy, CSS)：当路径冲突发生时，根据冲突的类型尝试对应的处理方案，若经判定冲突无法解决，将终止相应 AGV。详细描述请见附录 2。本文也考虑周期性重规划的策略，即每隔固定的时间对当前时钟进行同步，并对所有工作 AGV 进行一次路径重规划。本文在周期性重规划时若一辆工作 AGV 还在前往一个运输任务装货点的路上，则对于该运输任务使用任意一种任务驱动的指派策略，重新指派一辆 AGV。然后对所有工作 AGV 使用任意一种路径规划算法执行路径(重)规划。改进后的策略变成了周期性重调度(任务指派加路径规划)。由于本文考虑的 AGV 系统在任意时刻只能知道 AGV 正处于哪条通道或哪个节点但不知道它在该通道或节点上的具体位置，因此在路径重规划时，需要取 AGV 当前位置的完整行驶时间的期望值作为路径起点行驶时间，这可能会造成 AGV 额外的主动等待。在两次重调度之间，AGV 仍旧有可能发生冲突。如果是工作 AGV 活锁，则仍使用 CSS 策略中的处理方案；如果是其他冲突，则令相关 AGV 暂停直到重调度的时刻。如果工作 AGV 在路径(重)规划时找不到可行路径，则使用 CSS 策略中 AGV 终止与任务失败的处理方案。本文称上述冲突应对策略为冲突避免型策略(conflict-avoiding strategy, CAS)。此外，本文还考虑了 CSS 策略与 CAS 策略的集成，称之为混合型策略(mixed strategy, MS)。它既执行 CAS 中的周期性重调度，又对发生的路径冲突采用 CSS 策略即时进行处理。

一个完整的 AGV 系统调度算法，可由表 1 中任意一个任务指派策略、任意一个路径规划算法以及任意一个冲突应对策略组合而成。在不确定情景下，对于上述任意一个完整的调度算法，都

难以从理论上分析和计算刻画它们的效果与特性，因为路径冲突的发生(进而冲突应对策略的实施)具有随机性，且对应的概率分布难以计算。当前主流的仿真软件均无法直接实现对包含路径冲突应对策略在内的完整 AGV 调度算法进行建模与仿真。本文基于离散事件仿真技术，专门设计并实现了一个 AGV 系统仿真平台，可以对上述任意一个完整的调度算法进行全面深入的仿真评估。

表 1 完整 AGV 调度算法

Table 1 Completed AGV scheduling algorithm	
调度算法组成	调度算法种类
任务指派策略	NVF+NMF、NVF+EMF、LIVF+NMF、LIVF+EMF
路径规划算法	TWA、ASA
冲突应对策略	CSS、CAS、MS

## 3 AGV 系统仿真平台

本文基于离散事件仿真技术设计了一个 AGV 系统仿真平台。本仿真平台的设计思路是让 AGV 系统执行表 1 中的任务指派和路径规划算法，按照离散事件仿真的原理在离散时间点上更新系统状态，通过将路径冲突归类和将 AGV 活动细分，可以在推进仿真时间之前先判断在进入到下一个时间点的过程中是否存在路径冲突，若存在则执行表 1 中的冲突应对策略，改变原定的后续事件。整个 AGV 系统仿真平台如图 2 所示。系统调度模块主要包含运输任务的接受和指派、路径规划、冲突应对、AGV 终止与失败任务处理等；离散事件仿真模块在系统调度模块的驱动下，推进系统仿真时间，模拟各种离散事件的发生；GUI 模块主要包含数据输入输出、AGV 系统运行状态显示、仿真模式选择等。该仿真平台所模拟的 AGV 系统，除了满足第 2 节中所列假设以外，还满足以下假设：①AGV 可以连续地工作且不发生故障；②AGV 完成运输任务后便在原地等待新的运输任务；③运输任务没有硬时效约束，不会因为推迟而失败，只会由于路径冲突而判定为失败；④调度算法的计算耗时远远小于运输任务执行时



间, 因此, 调度算法的计算时间在仿真中忽略不记。

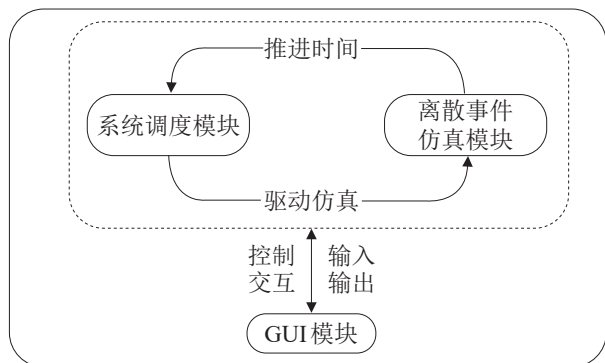


图 2 AGV 系统仿真平台架构  
Fig. 2 Architecture of AGV system simulation platform

### 3.1 系统调度模块

#### 3.1.1 任务指派逻辑

在任务指派策略方面, 系统调度模块允许指定使用表 1 中所列的 4 种组合任务指派策略中的任意一种。当某个生成的运输任务的到来时刻晚于当前系统时间时, 系统调度模块会驱动离散仿真模块运行, 推进系统时间直到运输任务到来的时刻。在仿真到任务到来时刻, 如果有空闲 AGV, 则采用指定的任务驱动的指派策略(NVF 或 LIVF)选择 AGV; 如果 AGV 均在执行其他运输任务, 则将该任务缓存至任务池, 并驱动离散仿真模块推进仿真直到系统中出现空闲 AGV。在此期间到来的运输任务也会被缓存至任务池中。当有 AGV 空闲时, 采用指定的车辆驱动的指派策略(NMF 或 EMF)选择运输任务。具体的任务指派流程如图 3 所示。

#### 3.1.2 路径规划逻辑

在路径规划算法方面, 系统调度模块允许指定使用表 1 中所列的 2 种路径规划算法中的任意一种。当 AGV 被分配了运输任务  $\{o, d, t\}$  时, 需要规划从 AGV 当前位置到  $o$  的取货路径与从  $o$  到  $d$  的送货路径, 2 条路径连接后得到总的任务路径。其中, AGV 在节点  $o$  上需要花费额外的装货时间, 在节点  $d$  上需要花费额外的卸货时间。在规划取

货路径或送货路径的时候, 无论使用哪一种算法, 都有可能因为节点被障碍占用而找不到可行路径。此时, 该任务就被视为暂不可执行, 返回至任务池, 一段时间后再进行判断。具体的路径规划流程如图 4 所示。

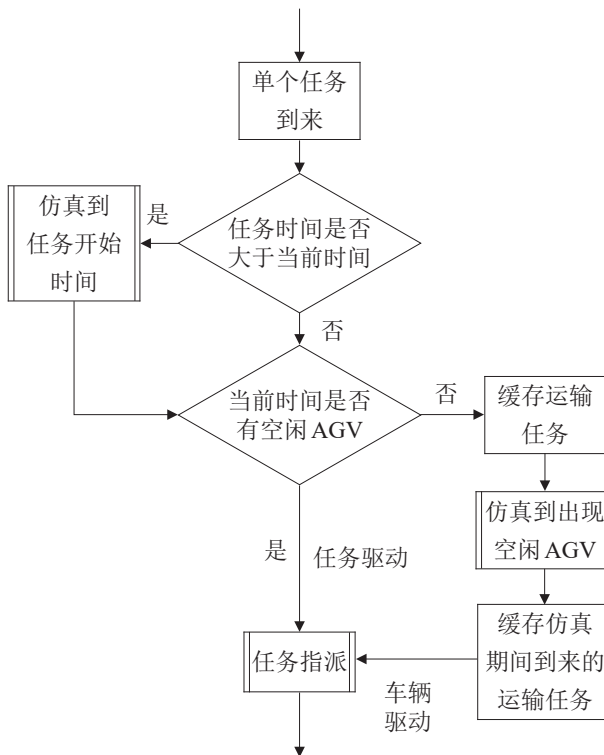


图 3 任务指派流程  
Fig. 3 Process of dispatching

#### 3.1.3 冲突应对逻辑

在冲突应对策略方面, 系统调度模块允许指定使用表 1 中所列的三种冲突应对策略中的任意一种。其中, CAS 中的周期性重调度功能依托于离散事件仿真模块中专门为此定义的事件而实现, 所有冲突应对策略中关于路径冲突的处理方案, 均当离散事件仿真模块中识别到冲突并判断出冲突类别后被唤起执行。

### 3.2 离散事件仿真模块

离散事件仿真模块在系统调度模块的驱动下, 对不确定情景下的 AGV 系统进行模拟。其中, AGV 运行时间的不确定性由为 AGV 通过节点所

需标称时间加上一个随机产生的偏差项实现，随机出现的障碍则通过随机地选择 $G(N', A')$ 中的节点并将其设置为不可用进行模拟。离散事件仿真模块的核心是离散事件的定义和系统仿真时间的推进。

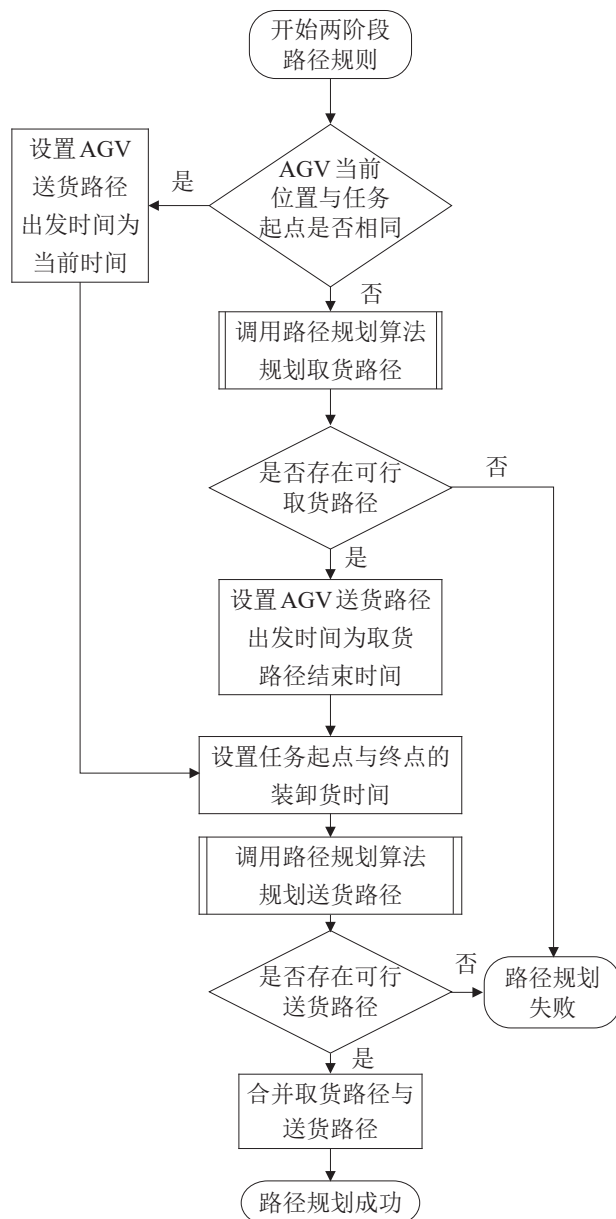


图4 路径规划流程  
Fig. 4 Process of routing

### 3.2.1 离散事件

在离散仿真中，仿真时钟按照一系列离散事件发生的时刻依次向前推进。本离散事件仿真模

块中最主要的事件为AGV活动事件，可细分为路径开始、位置改变、路径结束3类。AGV可视为在网络 $G(N', A')$ 上移动，每个节点需要一定的通过时间，而弧上没有通过时间。因此，在位置改变事件发生时，AGV刚离开上一个节点，将要开始进入另一个节点，此时AGV的位置存在歧义。为确保系统状态的唯一性以及正确判断，约定在位置改变事件发生时，AGV的位置为上一个节点。路径开始和路径结束这2个事件发生时AGV的位置无歧义。AGV完成整条路径过程中涉及的离散事件及其对应的位置，如图5所示。位置改变事件发生时，若AGV的下一位置正被障碍或其他AGV占用，路径冲突便发生，此时便会调用系统调度模块中的冲突应对策略进行处理，改变原定的AGV后续事件。

除了AGV活动事件之外，还有其他一些离散事件需要考虑。一是障碍事件，包括随机障碍物(非AGV)在节点处的出现与消失。为简便起见，约定当障碍随机地出现在一个节点上时，若已有AGV处于这个节点，那么它可正常离开，之后所有AGV无法进入该节点，直至障碍消失。此外，当系统调度模块中的冲突应对策略设定为CAS或MS时，由于需要周期性地对AGV进行重调度(时间间隔由参数决定)，因此，重调度事件也是需要专门考虑的离散事件。

上述离散事件可能在同一时刻发生，为确保仿真的有序推进，需要为其定义不同的优先级，如表2所示。其中，由于AGV活动事件中的3个子类同时发生的概率为0，因此，相互之间无需定义优先顺序。此外，对于障碍事件和AGV活动事件中的同一个子类，也可能存在同一时刻发生多个的情况，但是除了AGV位置改变事件之外，其余事件处理的优先顺序对仿真结果无影响，因此，在实际中按随机的顺序处理即可。而对于多个同时发生的AGV位置改变事件，如果AGV的下一个位置相同，则按照它们到达当前节点的先后顺序依次处理。

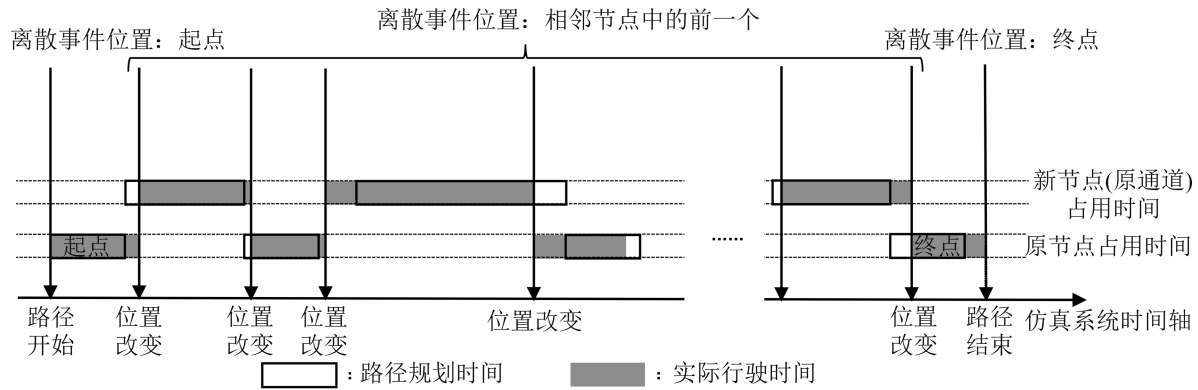


图5 AGV完成整条路径过程中的离散事件及其对应位置

Fig. 5 Discrete events and their corresponding positions during process that AGV finishes entire route

表2 离散事件优先级

Table 2 Priorities of discrete events

离散事件类型		优先顺序
障碍事件	障碍出现	1
	障碍消失	2
AGV 活动事件	路径开始	3
	路径结束	
	位置改变	
重调度事件		4

### 3.2.2 仿真推进逻辑

离散事件仿真模块根据离散事件发生的时刻与优先级, 以及系统在这些时刻的状态, 依次模拟各离散事件的发生, 并推进系统仿真时间。其中, 对于障碍事件, 通过对AGV路径网络的修改与重置实现对障碍出现和消失的模拟; 对于重调度事件, 通过调用系统调度模块中的任务指派和路径规划功能, 完成取货任务重指派和工作AGV路径重规划。对于AGV活动事件中的路径开始与结束事件, 由于不涉及AGV位置的改变, 不会触发路径冲突, 只需进行数据记录即可。对于路径开始事件, 还需依据已规划路径(基于通过节点的标称时间)与实际通过节点所需时间(已加上随机偏差)计算后续离散事件的发生时刻。对于AGV活动事件中的位置改变事件, 要进行AGV路径冲突判断, 如果没有发生路径冲突, 则实现AGV移动, 并且使用与路径开始事件相同的方式计算后续离散事件的发生时刻; 如果发生了路径冲突,

则先判断冲突类别, 然后调用系统调度模块中的冲突应对策略进行处理。

离散事件仿真模块推进系统时间的方式有3种: 仿真到出现空闲AGV、仿真到指定时间、仿真到所有任务结束。这3种方式对应了3种中止仿真的条件。对于仿真到出现空闲AGV的需求, 可以在离散事件完成后直接判断决定是否中止仿真。对于另外2种需求, 首先需要在离散事件完成后, 获取下一离散事件的发生时间。对于仿真到特定时间的需求, 如果该时刻在两个离散事件点之间, 则先将系统时间更新到指定时间然后中止仿真; 如果指定时间与下一事件发生时刻相同, 则先完成下一事件再中止仿真。对于仿真到所有任务结束的需求, 可以通过判断下一离散事件的发生时间是否为无穷大决定是否结束仿真。具体仿真推进与中止的逻辑如图6所示。

### 3.3 GUI模块

GUI模块是基于Python中的Tkinter 8.6开发的, 主要用于读取输入数据, 包括系统参数、调度算法等, 实时显示AGV系统的运行状态, 选择仿真模式以及输出仿真结果。整个GUI界面如图7所示。GUI界面左侧以2D俯视图形式显示AGV路径网络和AGV的实时位置与状态。由于每条路径的长度可以不同, 路径网络既可以采用图7中的标准化显示(以等长的边显示不同长度的通道),

也可以采用图8所示的真实比例显示(在设置文件中指定),以满足不同需求。路径网络上红色方块表示随机出现在节点或通道的障碍。黄色的多边形表示AGV,尖的一端是AGV的车头方向,上面的数字为AGV的编号。虚线框表示该AGV处于空闲状态,实线框表示该AGV处于工作状态,其中,黑色线框表示没有负载,绿色线框表示有负载。GUI界面左下角显示了当前系统仿真时间和当前任务池中以及未到来的任务数量。底部4个按钮用于仿真的模式选择与控制:步进仿真按钮每点击一次,仿真向前推进一步,并伴随相应AGV或路径网络状

态的显示更新;点击自动仿真按钮,则仿真持续自动向前推进且伴随动画显示,直至全部任务执行完毕,中途可以点击仿真暂停按钮退出自动仿真模式;点击直达仿真按钮,则在关闭动画显示的情况下以最快速度推进仿真直至全部任务执行完毕。GUI界面右侧以文字显示调度信息(任务指派结果、路径规划情况、路径冲突与解除情况等)、障碍信息(出现与消失)、AGV运行信息(包括路径开始、路径结束与到达离开节点等)。借助该可视化界面与相应的仿真控制按钮,可以直观地观察AGV的运行状态和整个调度算法的逻辑与表现。

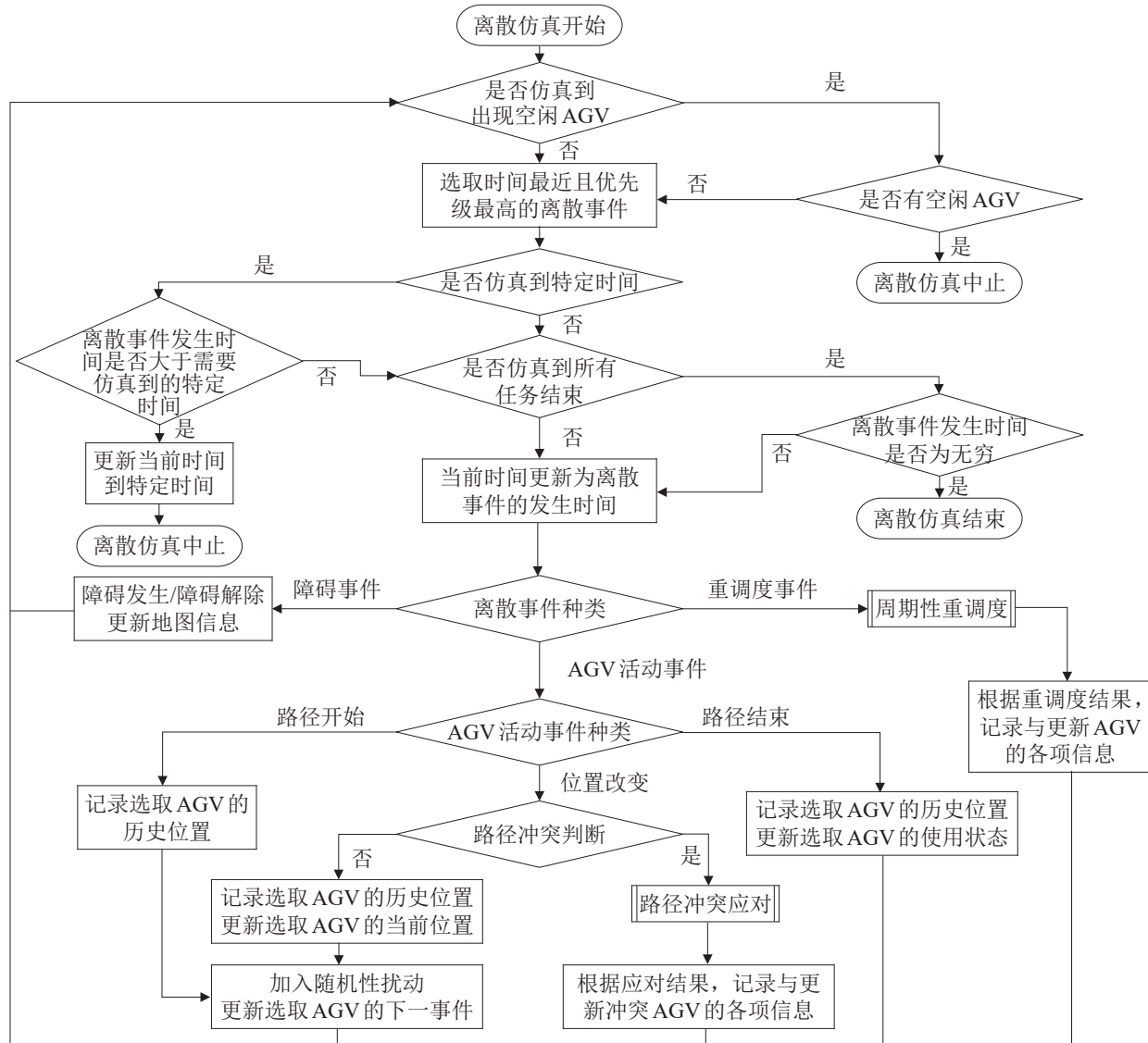


图6 系统仿真时间推进流程

Fig. 6 Process of driving system simulation clock

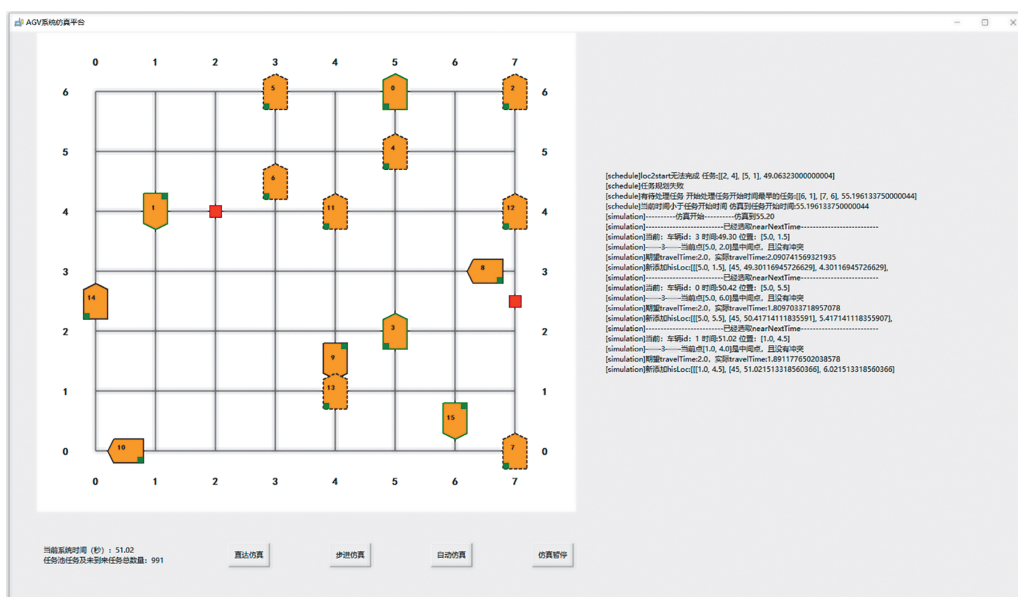


图 7 GUI 界面  
Fig. 7 GUI Interface

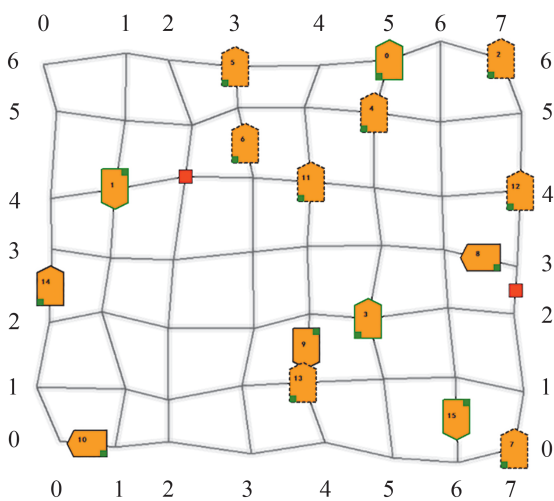


图 8 实验所用路径网络  
Fig. 8 Path network used in experiments

仿真平台输入信息众多, 为简便均通过读取同一目录下的 json 格式输入文件实现, 包括路径网络布局 and 显示模式、运输任务、AGV 数量、路径时间不确定性、随机障碍、调度算法及其参数、随机种子等。当设定的运输任务全部执行完毕时, 仿真自动终止, 并且在同一目录下生成 csv 格式输出文件, 包含基本实验信息和用于评价调度效果的统计量(系统总运行时间、AGV 总工作时间、任

务失败率等)。此外, GUI 界面右侧的文字信息会以 log 格式文件保存在同一目录下。

### 4 AGV 系统调度算法仿真实验

本节通过在 AGV 系统仿真平台上进行一系列实验, 展示该仿真平台的使用方法和价值。本节中的实验旨在探索仿真平台对 AGV 调度算法的评估与比较的能力, 而不是试图选出最优的调度算法。实际上, 对于一个给定的情景, 借助仿真平台寻找最优调度算法及其参数设定的问题, 是一个典型的仿真优化问题, 可借助该领域内的许多算法进行求解<sup>[31-32]</sup>。此外, 从实验结果可知, 在不同情景下最优的调度算法及其参数设定是不同的, 存在情景与最优调度之间的映射关系。因此也可以将该优化问题视作一个带协变量的仿真优化问题进行建模与求解<sup>[33-34]</sup>。

#### 4.1 基本实验设定与指标

实验考虑一个 7×8 网格布局的双向单通道 AGV 路径网络  $G(N, A)$ , 但通道并不是等长的, 添加了随机的扰动。如无特别说明, 本节中所说的随机分布或生成均特指在指定空间上的随机均匀

分布。路径网络的具体实现方式是，先令所有通道长度为10 m，随后为每个节点 $n \in N$ 横向和纵向分别施加一个 $[-5, 5]$ 的随机扰动，从而形成了如图8所示的路径网络。AGV总数为 $m$ 辆，在系统开始运行时，各AGV随机处于各节点 $n \in N$ 上且均为空闲状态，路径网络中所有通道和节点都可用。令AGV标称行驶速度 $v=1$  m/s，完整通过一个节点的时间期望 $2\delta=2$  s，对于通道 $a \in A$ ，AGV的通过时间期望为 $(l_a - 2)$  s。假设通过节点 $n \in N$ 时间的随机偏差 $\varepsilon_n \sim N(0, \theta \times 4)$ ，通过通道 $a \in A$ 时间的随机偏差 $\zeta_a \sim N(0, \theta(l_a - 2)^2)$ ， $\theta$ 决定了行驶速度/时间的不确定性的。实际随机生成的行驶时间还需要对0做截断处理。考虑1 000个运输任务，它们的起点与终点在 $N$ 上随机生成，且不相同。这些任务在时间段 $[0, T]$ 均匀地到来，其中， $T$ 决定了任务到来的密集程度，它由任务稀疏参数 $o$ 决定。令 $T_{\text{move}}^{\text{ideal}}$ 表示在忽略所有不确定性以及路径冲突情况下全部任务的最短行驶时间(可由ASA直接计算)的总和，则 $o$ 被定义为 $o = mT/T_{\text{move}}^{\text{ideal}}$ 。因此， $o$ 越大表示任务的到来越稀疏。AGV的装货操作时间 $\delta_1=2$  s，卸货操作时间 $\delta_2=2$  s。在路径网络的每个位置上，包括节点和通道，可以随机地生成障碍，这些障碍在300 s后自动消失。令 $b$ 表示网络中同时存在的障碍数量最大值，则每隔 $300/b$  s生成一个随机障碍。令 $e$ 表示网络中总的位置数量(本节 $e=153$ )，则可用 $c=b/e$ 表示障碍发生的频繁程度，它也是一个可变参数。若令 $c=0$ 则表示网络中不会出现随机障碍。

为了对完整的调度算法进行定量评估，考虑了3个指标：①系统总运行时间，即系统完成全部运输任务所需时间，记为 $T_{\text{run}}$ ；②AGV总工作时间，即所有AGV处于非空闲状态的时间总和，记为 $T_{\text{work}}$ ；③任务失败率，即由于路径冲突导致任务被记为失败的数量与总任务数量之间的比值，记为 $r$ 。显然， $r$ 的增大会使 $T_{\text{run}}$ 和 $T_{\text{work}}$ 减小，因此，为了更加客观的评估，本节中默认使用修正

后的系统总运行时间 $T_{\text{run}}/(1-r)$ 和AGV总工作时间 $T_{\text{work}}/(1-r)$ ，以下简称总运行时间和总工作时间。在每一个实验设定下，让AGV系统仿真平台重复运行1 000次，每次重复中AGV行驶时间的偏差、障碍出现的位置都重新生成，最终汇报平均的总运行时间、总工作时间和任务失败率。

基准实验设定： $m=16$ ， $\theta=0.15$ ， $o=2$ ， $c=0.04$ 。文献[14]中的实验表明NVF与NMF策略性能较其他任务指派策略更优，因此，在基准实验设定中任务指派策略采用NVF+NMF，路径规划算法采用TWA，冲突应对策略待定。

## 4.2 CAS中重调度周期的影响

对于CAS冲突应对策略，重调度周期是一个可变的参数，而且对它的最终表现有影响。重调度周期较短时，可以更好地避免路径冲突的发生，但会让AGV额外的主动等待时间变得更多。因此，在比较包含不同的冲突应对策略的调度算法之前，需要先为CAS设置一个合理的重调度周期。基于基准实验设定，令冲突应对策略为CAS，比较了在3种不同情况下，各调度效果指标随重调度周期长度增大的变化情况，仿真实验结果如图9所示，图中Case 1：基准实验设定；Case 2： $m=6$ ；Case 3： $o=5$ 。

以Case 1作为基准，Case 2中的AGV数量减少了，总运行时间便增加了，但节点时间窗占用减少了，使AGV可以走更短的路径，因此，总工作时间便减少了，并且由于重调度时找不到可行路径而导致的任务失败数量也减少了。Case 3相较Case 1，任务的到来变得更加稀疏了，因此，总运行时间必然增加，而且由于同时工作的AGV数量有所减少，任务失败率也相应降低一些，而总工作时间反而增大，是由于此时空闲AGV较多，而CAS在做重调度的时候无法将暂未引发冲突的空闲AGV考虑进来，导致使用TWA规划出的路径较易再与其他空闲AGV出现冲突，从而导致较长的工作时间。

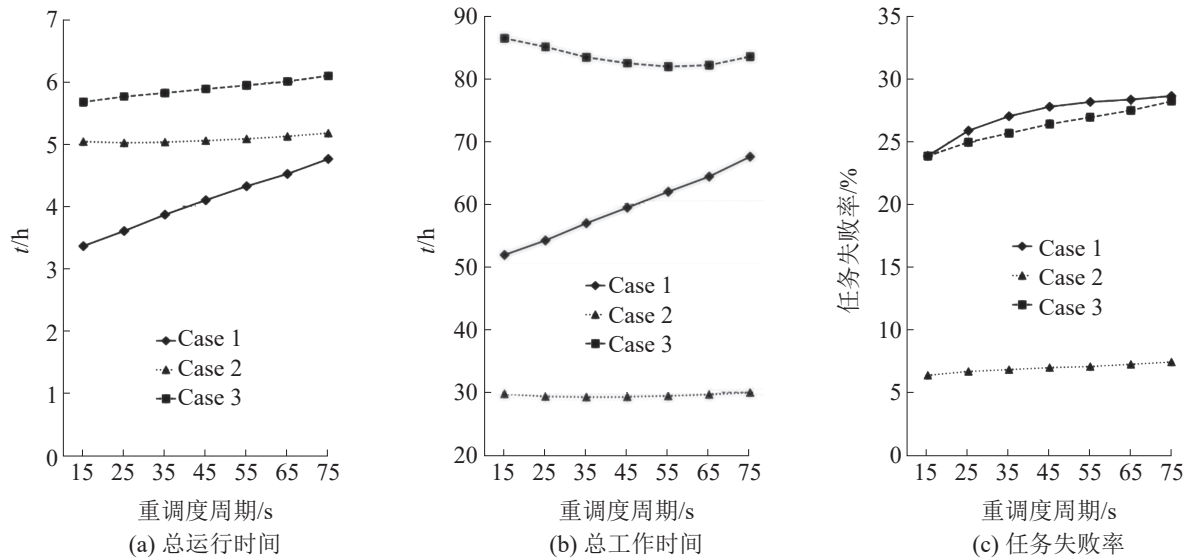


图9 不同情况下调度效果与重调度周期长度的关系

Fig. 9 Relationships between scheduling performance and rescheduling cycle length in different cases

在基准的 Case 1 中可以看到, 由于工作 AGV 较为拥挤冲突较多, 此时重调度周期长度越短调度效果越好, 因此, 重调度减少冲突的正面作用显著超过引发 AGV 主动等待的负面作用。而在 AGV 数量较少的 Case 2 中, 由于本身冲突较少, 因此, 重调度的正面作用与反面作用程度相当, 导致调度效果受重调度周期影响较小。在任务较稀疏的 Case 3 中, 反映调度效果的总运行时间和任务失败率, 与重调度周期的变化关系与 Case 1 中一致, 但总工作时间却在重调度周期长度小于 55 s 时随着其减小反而增加。通过进一步察看仿真平台输出的统计数据可以发现, 当重调度周期长度过小时, 虽然总的路径冲突数量减少, 但是空闲 AGV 冲突数量变多了, 反而导致更长的工作时间。

### 4.3 AGV 调度算法效果对比

为验证本文的 AGV 仿真平台评估和揭示不同 AGV 调度算法之间差异的能力。基于基准实验设定, 考虑在 3 种不同情况(Case 1: 基准实验设定; Case 2:  $c=0.08$ ; Case 3: 路径规划采用 ASA)下, 均尝试 CSS、CAS、MS 冲突应对策略。在每一种情况下, 调度算法仅在冲突应对策略上有差别, Case 3 与 Case 1 的对比可以观察路径规划算法对

调度效果的影响。实验结果如图 10 所示。

可以看到, 在基准的 Case 1 中, 使用 CSS 或 MS 时调度算法的效果在 3 项指标上都显著优于 CAS 的效果, 而 CSS 与 MS 之间没有太大差别。对于障碍发生频繁程度更高的 Case 2, 使用 MS 时调度算法的效果显著地优于使用其他冲突应对策略时的效果, 而 CSS 和 CAS 在不同指标上各有优劣。这是由于 3 种冲突应对策略内在逻辑不同导致的。CSS 中的冲突解除每次只能针对单个路径冲突, 至多涉及一辆 AGV 的路径重规划, 而 CAS 在做重调度的时候会对所有工作 AGV 进行路径重规划。当障碍发生频繁程度更高时, 路径网络信息变化更频繁, CSS 中需要不停地为出现障碍冲突的 AGV 进行路径重规划, 不如 CAS 中对所有工作 AGV 进行路径重规划高效, 从而表现在总运行时间上的结果是 CSS 需时更长。至于总工作时间, 由于 CAS 会引发 AGV 主动等待, 导致使用 CAS 时总工作时间更长, 而 MS 由于兼具 CSS 和 CAS 的优点, 因此表现最好。在使用了 ASA 路径规划算法的 Case 3 中, 3 种冲突应对策略间的对比与使用了 TWA 的 Case 1 中相同。总体来说, CAS 的表现差于 CSS 和 MS, 而 MS 略微优于 CSS。

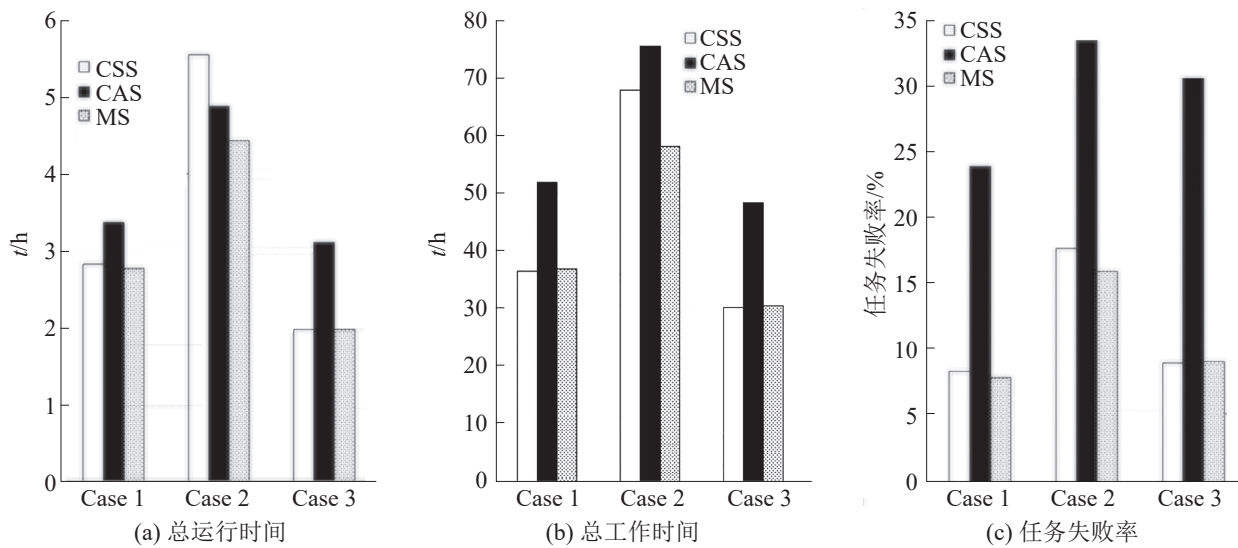


图 10 不同情况下几种调度算法的调度效果对比

Fig. 10 Performance comparison for several scheduling algorithms in different cases

对 Case 1 与 Case 3 做横向对比，观察路径规划算法对调度效果的影响。可以发现，对于基准实验设定中的任务密集度和障碍频繁程度，将路径规划算法从 TWA 变为 ASA，任务失败率有所增加，但是总运行时间和总工作时间却明显降低。在确定性路径时间假设下，TWA 优于 ASA，因为后者常常会规划出具有冲突的路径，这也正是导致上述失败率增加的原因。然而，当路径时间具有不确定之后，基于确定性时间窗假设的 TWA 规划出来的路径并不一定更好，因为一方面 AGV 为了避免冲突而走了路程较长的路径，另一方面由于路径时间的不确定性冲突还是无法避免，两者叠加之后可能还不如直接使用 ASA 而忽略可能存在的冲突。

#### 4.4 CSS 特性探究

以 CSS 为代表验证本文的 AGV 仿真平台对冲突应对策略的特性刻画能力。基于基准实验设定，令冲突应对策略为 CSS，并沿 5 个维度探究 CSS 的特性。AGV 数量维度： $m \in \{6, 7, \dots, 28\}$ ；时间不确定性维度： $\theta \in [0.1, 0.3]$ ；任务稀疏程度维度： $o \in [0, 5]$ ；障碍频繁程度维度： $c \in [0, 0.1]$ ；任务指派与路径规划维度：任务指派策略  $\in \{NVF + NMF,$

$NVF + EMF, LIVF + NMF, LIVF + EMF\}$ ，路径规划算法  $\in \{TWA, ASA\}$ 。

使用 CSS 冲突应对策略时的调度效果随着 AGV 数量增大的变化如图 11 所示。随着 AGV 数量增加，总运行时间先下降再上升。这是因为 AGV 数量的提升，一方面带来了运输能力的提升，使总运行时间减少；另一方面增加了路径冲突，使总运行时间增加。当 AGV 数量较少时，前者效应更加明显，而当 AGV 数量较多时，则相反。随着 AGV 数量增加，总工作时间和任务失败率均呈现大幅单调上升。这是因为路径冲突增加了，AGV 需要绕路以解除冲突，并且更加可能由于找不到可行路径而造成任务失败。从这个实验可以看到，由于路径冲突的存在，AGV 的数量并不是越多越好，而应在一个合理的范围。

使用 CSS 冲突应对策略时的调度效果随着 AGV 行驶时间不确定性增大的变化如图 12 所示。不确定性参数  $\theta$  值越大，表示 AGV 在路径(包括通道和节点)上行驶的时候，实际用时与期望用时之间偏差的随机性越大。可以看到，当不确定性变大时，各项指标均会轻微变差，这是因为 AGV 实际对路径的占用偏离 TWA 所规划的时间窗的程度变大，从而导致了更多路径冲突的发生。



使用 CSS 冲突应对策略时的调度效果随着任务稀疏程度增大的变化如图 13 所示。任务稀疏参数  $\theta$  越大, 表示任务的到来越稀疏。可以看到, 随着任务到来变得更加稀疏, 总运行时间上升, 并且越来越接近最后一个运输任务的到来时间, 这是因为任务需要在任务池中等待的情况变少, 到来之后便可

以被立刻执行。同时, 总工作时间和任务失败率均有所上升, 这是因为空闲 AGV 数量变多时, 调度算法使用 TWA 规划路径时不会考虑空闲 AGV 对路径的占用, 导致规划出的路径较易出现空闲 AGV 冲突, 并且 CSS 在处理此类冲突时会让空闲 AGV 避让, 进一步增加了 AGV 的工作时间。

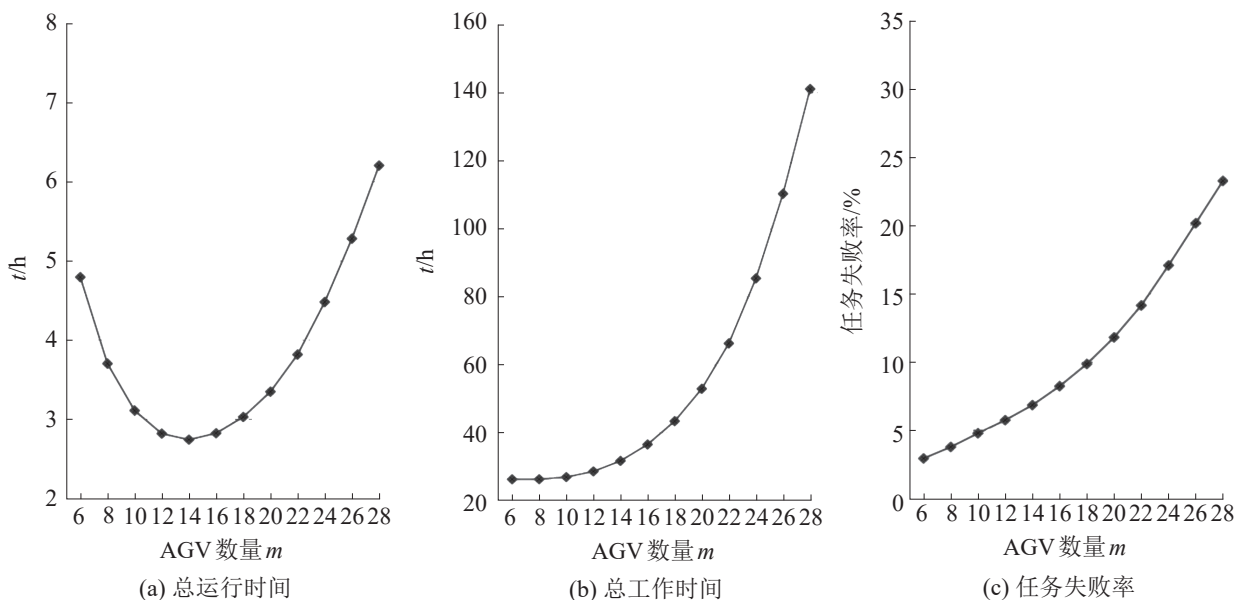


图 11 使用 CSS 的调度算法效果与 AGV 数量之间的关系  
Fig. 11 Performance of scheduling algorithm with CSS with respect to AGV number

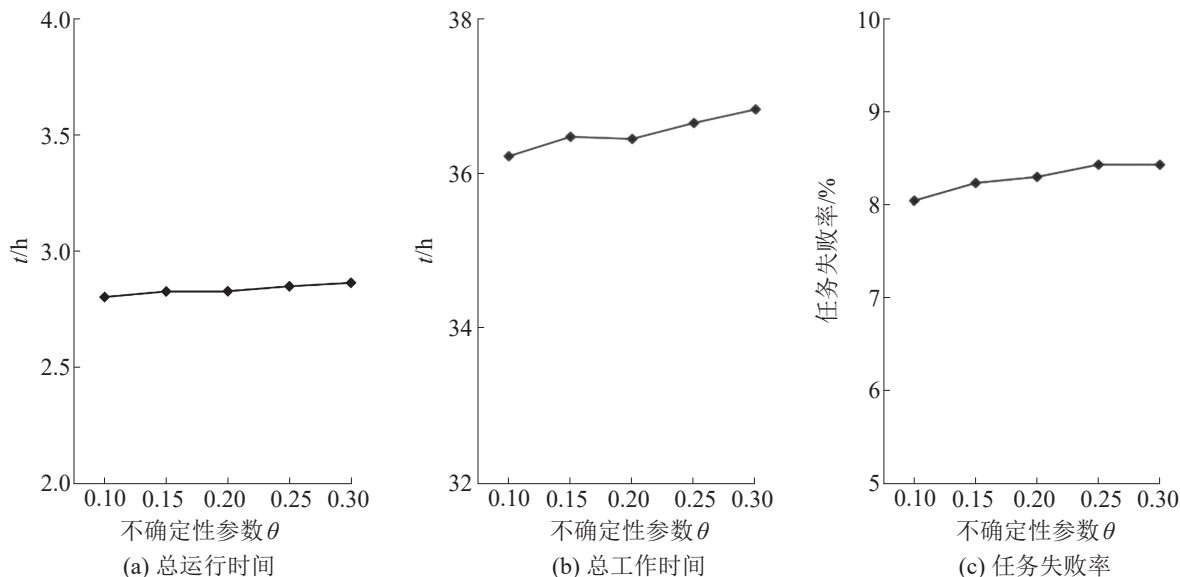


图 12 使用 CSS 的调度算法效果与 AGV 行驶时间不确定性之间的关系  
Fig. 12 Performance of scheduling algorithm with CSS with respect to uncertainty of AGV travel time

使用 CSS 冲突应对策略时的调度效果随着障碍频繁程度增大的变化如图 14 所示。障碍频繁参数  $c$  越大，表示出现在路径网络中的随机障碍数量越多。可以看到，随着障碍数量增加，各指标都大幅度变差，因为路径网络中可用位置变少了，

AGV 需要行驶更长的距离才能完成任务，也使系统总运行时间变长。另外，增加的障碍使障碍冲突的发生变多，也使路径规划或重规划时因无可行路径而导致的任务失败变多。

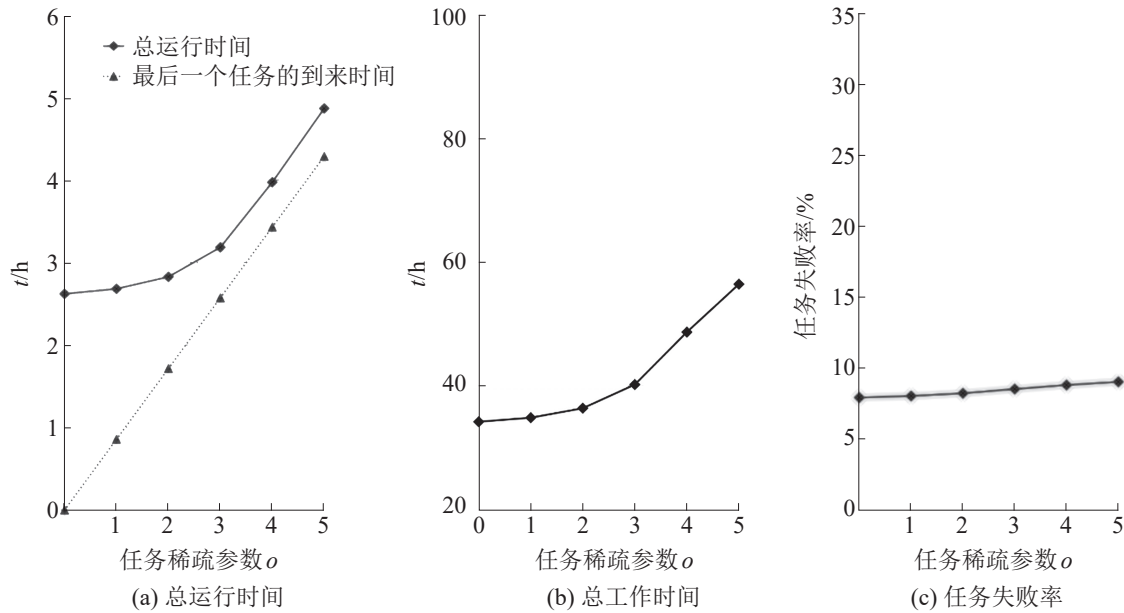


图 13 使用 CSS 的调度算法效果与任务稀疏程度之间的关系

Fig. 13 Performance of scheduling algorithm with CSS with respect to task sparsity

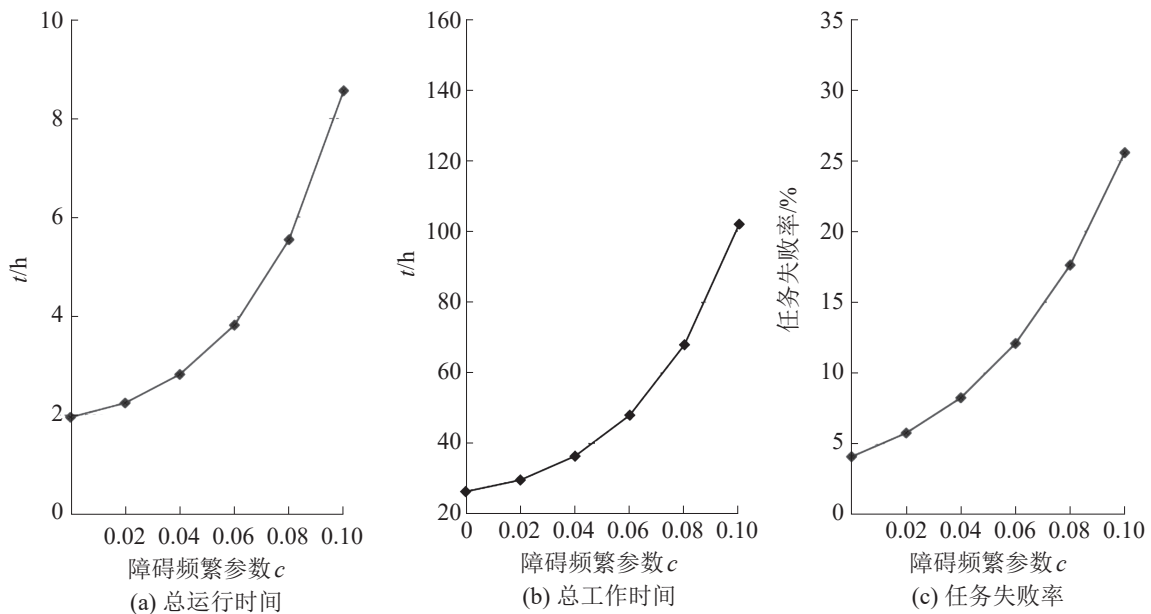


图 14 使用 CSS 的调度算法效果与障碍频繁程度之间的关系

Fig. 14 Performance of scheduling algorithm with CSS with respect to barrier frequency

使用CSS冲突应对策略时的调度效果随着任务指派与路径规划的变化如图15所示。可以看到对于这个具体的问题,在全部任务指派策略组合下使用了ASA的调度算法在总运行时间和总工作时间内相对TWA都有更好的表现,尤其当任务指派策略为NVF+EMF或LIVF+EMF时,ASA的优势更加明显,并且在任务失败率上表现也更优。当任务指派策略为NVF+NMF或LIVF+NMF时,使用了ASA的调度算法在任务失败率上表现略差

于TWA。总体上看,对于这个具体的问题,ASA比TWA具有更好的表现,其原因与4.2节中所阐述的相同。对比不同的任务指派策略可知,对于这个具体的问题,当采用TWA进行路径规划时,NVF+NMF与LIVF+NMF显著优于其他2个任务指派策略,这和Le-Anh等<sup>[14]</sup>使用真实工厂数据对任务指派策略进行比较时得出的结论相同。当采用ASA进行路径规划时,任务指派策略之间的差别相对较小。

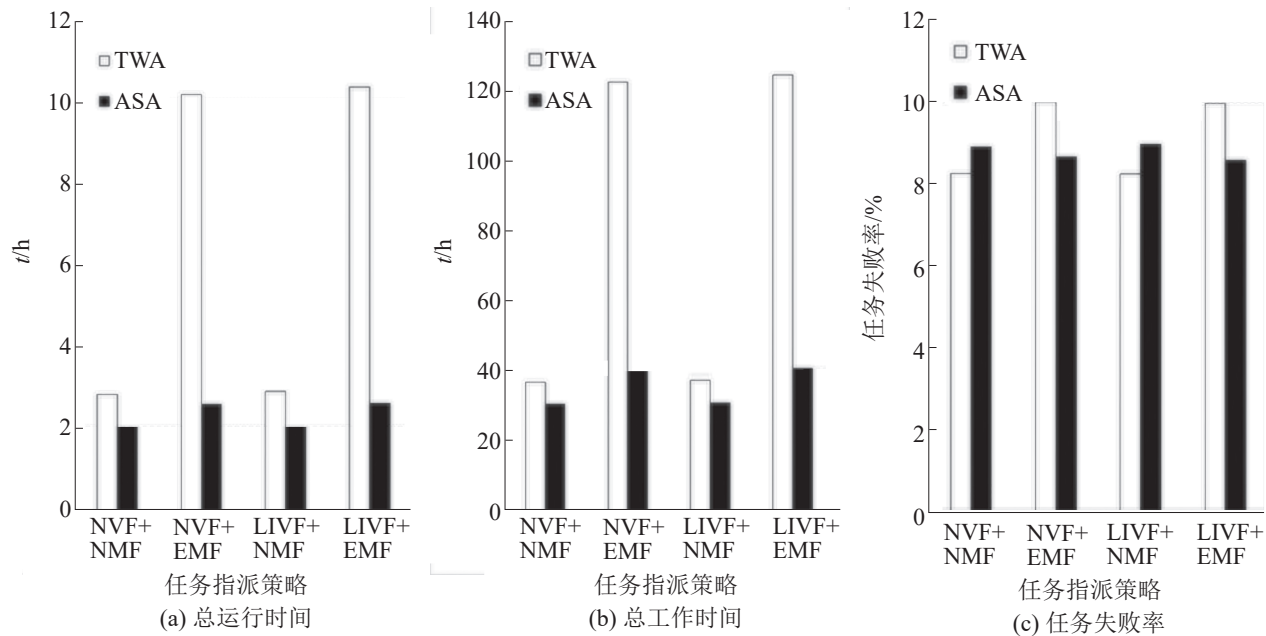


图15 使用CSS的调度算法效果与任务指派策略和路径规划算法之间的关系

Fig. 15 Performance of scheduling algorithm with CSS with respect to dispatching strategy and routing algorithm

本文设计的AGV系统仿真平台,能够很好地对一个调度算法进行评估,刻画它多方面的特性,也能够实现对不同调度算法表现的对比。此外,通过实验也可以发现,对于不同的不确定情景下的AGV调度问题,最优的调度算法(包含任务指派、路径规划与冲突应对)及其相应参数设置是不同的,或者不同的调度算法在不同的性能指标上各有优劣。而这些对比关系很难提前通过遍历全部的问题设定罗列出来,必须要借助仿真平台。对于一个具体的AGV调度问题,尝试比较不同的调度算法及其相应参数,从而针对关注的性能指

标选出最合适的调度算法,而这也正是本文设计开发这样一个AGV系统仿真平台的意义。

## 5 结论

本文着眼于不确定情景下的AGV系统调度问题,考虑到包含任务指派、路径规划与冲突应对的调度算法难以从理论上进行分析与评估,设计并开发了基于离散事件仿真技术的AGV系统仿真平台。该平台可以设置调度问题,选择调度算法,并且具有可视化界面,使用起来灵活便捷,能够准确有效地对不确定情景下的调度算法进行评估

与比较。基于该仿真平台，未来可以从仿真优化角度深入研究调度算法的优化问题。作为第一个版本，本文的AGV系统仿真平台仍有值得改进的地方，例如，增强可视化效果与人机交互，通过开放接口允许用户自定义调度算法等，均值得作为后续研究。

本文开发的AGV系统仿真平台连同展示的实验算例发布于<https://simopt.github.io/AGVSim>。

### 参考文献:

- [1] 佚名. 中华人民共和国国家标准-物流术语[J]. 交通建设与管理, 2007(10): 106-120.
- [2] Oyekanlu E A, Smith A C, Thomas W P, et al. A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-based Smart Manufacturing Applications[J]. IEEE Access, 2020, 8: 202312-202353.
- [3] Schneier M, Bostelman R. Literature Review of Mobile Robots for Manufacturing[M]. Gaithersburg: U. S. Department of Commerce, National Institute of Standards and Technology, 2015.
- [4] Iwasa M, Toda Y, Saputra A A, et al. Path Planning of the Autonomous Mobile Robot by Using Real-time Rolling Risk Estimation with Fuzzy Inference[C]//2017 IEEE Symposium Series on Computational Intelligence (SSCI). Piscataway, NJ, USA: IEEE, 2017: 1-6.
- [5] Anon. Automated Guided Vehicle Market Size, Share & Trends Analysis Report By Vehicle Type, By Navigation Technology, ApplicationBy, By End-use Industry, ComponentBy, By Battery Type, RegionBy, and Segment Forecasts, 2021-2028[EB/OL]. [2022-03-16]. <https://www.fortunebusinessinsights.com/automated-guided-vehicle-agv-market-101966>.
- [6] 现代物流. 运用数字孪生让AGV进发仓储活力[EB/OL]. (2021-12-13) [2022-04-13]. <http://www.logclub.com/articleInfo/NDE5MjE>.
- [7] 苏州艾吉威机器人有限公司.智慧物流管理系统[EB/OL]. [2022-04-13]. [http://www.agvsz.com/productclass\\_2](http://www.agvsz.com/productclass_2).
- [8] Anon. FUJI Taps Digital Twin for Simulation in Smart Factories | AEI[EB/OL]. (2021-07-06) [2022-04-13]. <https://aei.dempa.net/archives/345>.
- [9] Staff M M H. KION Group Starts Research Project ARIBIC Focused on Digital Twins of Warehouses[EB/OL]. (2021-08-04) [2022-04-13]. [https://www.scmr.com/article/kion\\_group\\_starts\\_research\\_project\\_aribic\\_focused\\_on\\_digital\\_twins\\_of\\_wareh](https://www.scmr.com/article/kion_group_starts_research_project_aribic_focused_on_digital_twins_of_wareh).
- [10] Mäntysaari J. Time Window Based Vehicle Routing System[D]. [S.l.]: Tampere University, 2017.
- [11] Luís M S Dias, António A C Vieira, Pereira G A B, et al. Discrete Simulation Software Ranking-A Top List of the Worldwide Most Popular and Used Tools[C]//2016 Winter Simulation Conference (WSC). Piscataway, NJ, USA: IEEE, 2016: 1060-1071.
- [12] 熊俊涛, 李中行, 林国茂, 等. 基于交互式控制的AGV实时仿真系统研究[J]. 系统仿真学报, 2021, 33(3): 657-668. Xiong Juntao, Li Zhongxing, Lin Guomao, et al. Research on AGV Simulation System Based on Interactive Control[J]. Journal of System Simulation, 2021, 33(3): 657-668.
- [13] Robinson S. Simulation: The Practice of Model Development and Use[M]. New York: Palgrave Macmillan, 2014.
- [14] René (M) B M de Koster, Le-Anh T, J Robert van der Meer. Testing and Classifying Vehicle Dispatching Rules in Three Real-world Settings[J]. Journal of Operations Management, 2004, 22(4): 369-386.
- [15] Vis I F A. Survey of Research in the Design and Control of Automated Guided Vehicle Systems[J]. European Journal of Operational Research, 2006, 170(3): 677-709.
- [16] Bartholdi J J III, Platzman L K. Decentralized Control of Automated Guided Vehicles on a Simple Loop[J]. IIE Transactions, 1989, 21(1): 76-81.
- [17] Egbelu P J, Tanchoco J M A. Characterization of Automatic Guided Vehicle Dispatching Rules[J]. International Journal of Production Research, 1984, 22(3): 359-374.
- [18] Qiu Ling, Hsu W J, Huang S Y, et al. Scheduling and Routing Algorithms for AGVs: A Survey[J]. International Journal of Production Research, 2002, 40(3): 745-760.
- [19] Rolf H Möhring, Ekkehard Köhler, Gawrilow E, et al. Conflict-free Real-time AGV Routing[C]//Operations research proceedings 2004. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 18-24.
- [20] Broadbent A J. Free Ranging AGV Systems: Promises, Problems and Pathways[C]//Proceedings of the 2nd International Conference on Automated Materials Handling. [S.l.]: [s.n.], 1985: 221-237.
- [21] Dijkstra E W. A Note on Two Problems in Connexion with Graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [22] Huang J, Palekar U S, Kapoor S G. A Labeling Algorithm for the Navigation of Automated Guided Vehicles[J]. Journal of Engineering for Industry, 1993, 115(3): 315-321.
- [23] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the

- Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [24] Taghaboni-Dutta F, Tanchoco J M A. Comparison of Dynamic Routing Techniques for Automated Guided Vehicle System[J]. International Journal of Production Research, 1995, 33(10): 2653-2669.
- [25] Sun X C, Tchernev N. Impact of Empty Vehicle Flow on Optimal Flow Path Design for Unidirectional AGV Systems[J]. International Journal of Production Research, 1996, 34(10): 2827-2852.
- [26] Malmberg C J. A Model for the Design of Zone Control Automated Guided Vehicle Systems[J]. International Journal of Production Research, 1990, 28(10): 1741-1758.
- [27] 付建林, 张恒志, 张剑, 等. 自动导引车调度优化研究综述[J]. 系统仿真学报, 2020, 32(9): 1664-1675.  
Fu Jianlin, Zhang Hengzhi, Zhang Jian, et al. Review on AGV Scheduling Optimization[J]. Journal of System Simulation, 2020, 32(9): 1664-1675.
- [28] Nishi T, Morinaka S, Konishi M. A Distributed Routing Method for AGVs Under Motion Delay Disturbance[J]. Robotics and Computer-Integrated Manufacturing, 2007, 23(5): 517-532.
- [29] 珍珍. 618发力! 京东“亚洲一号”对外开放, 真如外界传的那样壮观[EB/OL]. (2019-06-07) [2022-03-27]. <https://www.iyiyou.com/news/20190607102200>.
- [30] Xiang Xi, Liu Changchun, Miao Lixin. Storage Assignment and Order Batching Problem in Kiva Mobile Fulfilment System[J]. Engineering Optimization, 2018, 50(11): 1941-1962.
- [31] Hong L J, Nelson B L. A Brief Introduction to Optimization Via Simulation[C]//Proceedings of the 2009 Winter Simulation Conference (WSC). Piscataway, NJ, USA: IEEE, 2009: 75-85.
- [32] Fu M C. Handbook of Simulation Optimization[M]. New York: Springer New York, 2015.
- [33] Shen Haihui, Hong L J, Zhang Xiaowei. Ranking and Selection with Covariates for Personalized Decision Making[J]. INFORMS Journal on Computing, 2021, 33(4): 1500-1519.
- [34] Xiong Shifeng. Personalized Optimization and Its Implementation in Computer Experiments[J]. IIEE Transactions, 2020, 52(5): 528-536.

## 附录1: 时间窗无冲突最短路算法(TWA)细节

本文考虑的时间窗无冲突最短路算法(TWA), 是对转换后的只有节点时间的网络  $G(N', A')$  进行定义的。对于任意一个节点, 可以定义3类时间窗: 可用时间窗  $w^u = [t_1, t_4]$ ; 可到达时间窗  $w^a = [t_1, t_2]$ ; 可离开时间窗  $w^d =$

$[t_3, t_4]$ 。其中, 可离开时间窗仅在证明算法最优性时需要。可用时间窗表示节点空闲的时间。因为AGV通过这个节点需要时间, 例如, 记为  $\tau$ , 那么它必须要在  $t_4 - \tau$  之前到达这个节点准备进入, 即  $t_2 = t_4 - \tau$ 。显然, 当  $t_4 - t_1 < \tau$  时, 该AGV一定无法在这个时间窗内通过, 此时可到达时间窗的区间也会出现矛盾。对于一个节点, 会有若干个可用时间窗, 一个新规划的AGV必须在该节点的某个可用时间窗内完整地通过这个节点, 才不会与已规划好的其他AGV发生路径冲突。在初始状态下, 一个节点的可用时间窗与可到达时间窗皆为  $[0, \infty]$ 。假设已经规划了一辆AGV在  $[t, t + \tau]$  行驶通过这一节点, 那么节点的可用时间窗则变为  $[0, t] \cup [t + \tau, \infty]$ , 其可到达时间窗变为  $[0, t - \tau] \cup [t + \tau, \infty]$ 。对于  $n'_i \in G(N', A')$ , 它的可用时间窗和可到达时间窗可由若干段区间组成, 令  $w_{ik}^u = [t_{1ik}, t_{4ik}]$  和  $w_{ik}^a = [t_{1ik}, t_{2ik}]$  分别表示节点  $n'_i$  上的第  $k$  段可用时间窗和可到达时间窗。

假设在  $t$  时刻, 给定其他已有的规划, 要寻找一条由节点  $o$  至节点  $d$  的无冲突最短路  $r$ 。对于  $n'_i \in G(N', A')$ , 定义它的标签为  $label_i = \{n'_i, \alpha_i, \beta_i, \lambda_i, k_i\}$ , 其中  $i$  是标签的序号,  $n'_i$  是标签对应节点,  $\alpha_i$  是到达节点的时刻,  $\beta_i$  是最早可离开节点的时刻,  $\lambda_i$  是与它相连的上一个节点的标签序号,  $k_i$  是与  $\alpha_i$  和  $\beta_i$  对应的时间窗段的序号。此外, 对于  $n'_i \in G(N', A')$ , 令  $\tau_i$  为AGV通过该节点的期望时间, 即忽略随机偏差,  $\gamma_i$  为从  $n'_i$  到  $d$  不考虑路径冲突的最短路径时间,  $C_{n'_i}$  为与  $n'_i$  相邻节点的集合。定义函数  $f(n'_i) = \beta_i + \min_{j \in C_{n'_i}} \gamma_j$ 。那么, 无冲突最短路  $r$  可通过下列时间窗无冲突最短路算法产生。在Huang等提出的时间窗无冲突最短路算法中,  $f(n'_i) = \beta_i^{[22]}$ 。使用本文定义的  $f(n'_i)$  可以提高算法的搜索效率, 并且由于它依旧满足A\*算法的可行性条件, 因此, 仍然可以确保得出的无冲突路径的最优性<sup>[23]</sup>。

## 附录2: 冲突解决型策略(CSS)细节

### (1) AGV终止与任务失败

当某辆正在工作的AGV被终止时, 可分为3种情况处理: ①该AGV正在前往某个运输任务的装货点, 此时将相应任务返回至任务池; ②该AGV已经在运输途中, 此时将该任务记为失败并清空(在实际中代表着人工介入处理); ③该AGV并未被指派任务, 只是为了避让其他AGV, 此时无任务需要处理。在任一情况下, 该AGV在原地被重置为空闲, 清除该AGV原规划路径以及相应的节点占用时间窗, 已花费的时间计入至该AGV的工作时间中。

### (2) 障碍冲突的处理

清除该AGV原规划路径以及相应的节点占用时间窗, 将发生冲突的节点设置为不可用, 立即对AGV使用2.2节中任意一种路径规划算法执行路径重规划; 若无可行路径, 则终止该AGV。

### (3) 工作AGV活锁的处理

按照到来时间先后顺序(先看到引发冲突的节点的时

间,若相同则再看到达上一个节点的时间,以此类推),使AGV依次通过该节点。

#### (4) 工作AGV对头死锁的处理

假设当AGV1 ∈  $V_{\text{busy}}$  即将离开当前节点  $n'_1 \in N'$  进入规划路径的下一个节点  $n'_2 \in N'$  时,发现它正被AGV2 ∈  $V_{\text{busy}}$  占用且AGV2的规划路径的下一个节点为  $n'_1$ 。将  $n'_2$  临时设置为不可用,清除AGV1原规划路径以及相应节点占用时间窗,对AGV1使用2.2节中任意一种路径规划算法执行路径重规划,若能找到可行路径,则AGV2采用工作AGV活锁的处理方案进入  $n'_1$ ;若无可行路径,将  $n'_1$  临时设置为不可用,清除AGV2原规划路径以及相应节点占用时间窗,对AGV2使用2.2节中任意一种路径规划算法执行路径重规划,若能找到可行路径,则AGV1采用工作AGV活锁的处理方案进入  $n'_2$ ;否则,终止这两辆AGV。

#### (5) 工作AGV循环死锁的处理

当识别到引发冲突判断的2辆AGV是处于循环死锁中时,立即终止这2辆AGV。

#### (6) 空闲AGV冲突的处理

假设当AGV1 ∈  $V_{\text{busy}}$  即将进入规划路径的下一个节点  $n' \in N'$  时,发现它正被AGV2 ∈  $V_{\text{idle}}$  占用。首先判断AGV2是否可能暂时离开  $n'$ 。根据网络  $G(N', A')$  的结构以及当下各节点占用情况,为了避让AGV1,AGV2需要先移动到避让节点。避让节点有3种类型:第1类避让节点是与  $n'$  相连,不在AGV1原规划路径上,且当前未被占用的节

点;第2类避让节点是与  $n'$  相连,不在AGV1原规划路径上,但当前被其他AGV,如AGV3占用着;第3类避让节点是与  $n'$  相连但在AGV1原规划路径上。若存在第1类避让节点,则令AGV2进入该节点;若不存在第1类避让节点但存在第2类避让节点,则先规划AGV2进入该节点,根据AGV2与AGV3之间的冲突类型(可能是工作AGV活锁、工作AGV对头死锁、空闲AGV冲突中任意一种)采用相应的处理方案;若只有第3类避让节点可选,则令AGV2进入该节点。在上述3种情况下,AGV1与AGV2之间变成工作AGV活锁。如果AGV2进入的是第3类避让节点,那么之后还会发生空闲AGV冲突(AGV1的下一个节点被AGV2占用),需要再次采用空闲AGV冲突的处理方案;如果AGV2进入的是1类或2类避让节点,则需要AGV1离开之后返回  $n'$ 。如果上述3类避让节点均不存在,则将AGV2视为障碍,使用障碍冲突的处理方案。

#### (7) 处理方案之间形成的死循环

上述一系列的冲突处理方案足以成功应对绝大多数情况,但是还是存在极少数的情况,会使上述的方案陷入死循环。例如,空闲AGV在试图避让的过程中又与其他AGV产生了无法解决的冲突,导致其在原地又被重置为空闲AGV,此时死循环便出现了。本文使用设置循环次数上限的方法进行处理,如果一辆AGV在同一位置连续发生同一类型的路径冲突超过3次,则终止该AGV。