

苏州大学

本科毕业设计(论文)

学院(部)	数学科学学院		
题 目	智能仓库路径规划算法设计与实现		
年 级	2018 级	专业	数学基地
班 级	18 基地	学号	1807402006
姓 名	王蒋翼		
指导老师	丁睿	职称	教授
论文提交日期			

# 苏州大学

## 本科毕业设计（论文）独创性声明

本人郑重声明：所提交的本科毕业设计（论文）是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本设计（论文）不含其他个人或集体已经发表或撰写过的研究成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

作者签名： 王蒋翼

日 期： 2022.5.6

# 苏州大学

## 本科毕业设计（论文）使用授权声明

本人完全了解苏州大学关于收集、保存和使用本科毕业设计（论文）的规定，即：本科毕业设计（论文）的著作权以及文中研究成果的知识产权归属苏州大学。苏州大学有权向国家有关部门或第三方机构送交毕业设计（论文）的复印件和电子文档，允许毕业设计（论文）被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编毕业设计（论文），可以将毕业设计（论文）的全部或部分内容编入有关数据库进行检索。

涉密设计（论文）☐

本设计（论文）属 \_\_\_\_\_ 在 \_\_\_\_\_ 年 \_\_\_\_\_ 月解密后适用本规定。

非涉密设计（论文）☐

论文作者签名： 王蒋翼 日期： 2022.5.6

导师签名： 丁睿 日期： 2022.5.6

# 智能仓库路径规划算法设计与实现

## 摘要

随着经济社会的不断发展，人们对物流行业的需求不断提高。传统劳动力导向的物流行业日益无法满足人们的需要，因此，我们需要一种效率更高的物流模式。随着仓库规模的不断扩大，各大电商平台逐渐开始探索智能仓库这一未知领域，希望凭借自动化设备、高集成化系统来实现工作效率的飞跃。

移动机器人的路径规划问题，是智能仓库能否高效运作的重中之重。目前，关于智能仓库移动机器人的路径规划问题，众多研究集中在单一移动机器人的情形，但在多个移动机器人的无碰路径规划问题方面，目前的研究结果相对较少。本文基于前人的研究成果，给出一套完善的多移动机器人无碰路径规划的算法方案，并且通过仿真模拟实验验证了该方案的高效性与可靠性。

研究主要包括以下方面：

在通过对移动机器人以及智能仓库进行一系列合理假设的前提下，为智能仓库路径搜索问题确立起一个合适的数学模型，同时给出在这一模型下相关定义的严格刻画。

基于前人的相关研究，利用 $A^*$ 算法、 $\epsilon - greedy$ 搜索等相关已有方法，对传统蚁群算法概率选择机制、启发信息设置等多方面进行改进。通过避免算法初期的盲目搜索，在加快算法收敛速度的同时，提高了所得的解的质量。

根据贪心算法，将多移动机器人无碰路径搜索问题分解成一系列动态障碍的单一机器人路径搜索问题。在对所有可能碰撞类型进行详细分析的基础上，结合预约表方法以及改进后的蚁群算法对一系列子问题进行求解。最后，对移动机器人进行单行道策略设定，从而解决了所有可能的碰撞问题。换言之，理论上，此算法可为多移动机器人无碰路径搜索问题在较短计算时间内提供一个质量好的次最优解。

通过 Python 软件，对智能仓库系统的运行状态进行仿真模拟。模拟实验证明了多移动机器人调度算法的高效性与可靠性，可较好地避免多移动机器人之间可能存在的碰撞问题。

关键词：智能仓库、多小车路径规划、蚁群算法、贪心算法、 $A^*$ 算法

# Design and Implementation of Path-Planning Algorithms for Intelligent Warehouse

## Abstract

With the rapid development of economy and society, the demand for the logistics industry is increasing. The traditional labor-oriented logistics industry is unable to meet people's needs. Therefore, we need a more efficient logistics mode. With the expansion of warehouse scale, major e-commerce platforms gradually began to explore the unknown field of intelligent warehouse, hoping to achieve a leap in work efficiency by virtue of automatic equipment and highly integrated system.

The top priority of whether the intelligent warehouse can operate efficiently is the path planning problem of mobile robot. At present, many researches focus on the case of a single mobile robot, but there are relatively few research results on the collision-free path planning of multi-mobile robots. Based on the previous research results, this paper presents a good algorithm scheme for collision-free path planning of multi-mobile robots, and the simulation experiments verify the efficiency and reliability of the scheme.

The study included the following:

Based on a series of reasonable assumptions for mobile robots and intelligent warehouses, an appropriate mathematical model for the path search problem of intelligent warehouses is established. Moreover, we give a rigorous description of the relevant definitions under this model.

Based on previous studies, we draw lessons from  $A^*$  algorithm,  $\epsilon$ -greedy searching and other related existing methods in order to improve the probability selection mechanism and heuristic information setting of the traditional ant colony algorithm. By avoiding the blind search in the initial stage of the algorithm, the convergence speed of the algorithm is accelerated and the quality of the solution is guaranteed.

According to the greedy algorithm, the collision-free path search problem of multi-mobile robots is decomposed into a series of single robot path search problems with dynamic obstacles. Based on the detailed analysis of all possible collision types, this can be solved by combining the reservation table method and the improved ant colony algorithm. Eventually, the one-way road strategy is set for the mobile robot, so as to completely solve all possible collision problems. In other words, this algorithm can provide a sub-optimal solution with good quality for the collision-free path search problem of multiple mobile robots at small computational costs.

The intelligent warehouse system is simulated by Python software. Moreover, simulation experiments prove that the multi-mobile robot scheduling algorithm is efficient and reliable, and can avoid the possible collision problem between multiple mobile robots.

Keywords: intelligent warehouse, MAPF, Ant Colony Algorithm, greedy algorithm,  $A^*$  algorithm

## 目录

摘要.....	i
Abstract.....	ii
第 1 章 前言.....	1
1.1 研究背景与研究意义.....	1
1.2 路径规划的任务描述以及算法现状.....	2
1.3 论文主要工作.....	3
1.4 论文主要架构.....	3
第 2 章 智能仓库建模.....	5
2.1 智能仓库系统运作流程描述.....	5
2.2 基于栅格法的环境建模.....	6
2.3 移动机器人运动建模.....	7
2.4 智能仓库具体模型.....	8
第 3 章 路径规划任务分析.....	10
3.1 单小车路径规划问题分析.....	10
3.2 多小车路径规划问题分析.....	10
第 4 章 小车路径规划算法设计.....	12
4.1 单个小车的路径规划算法设计.....	12
4.1.1 传统蚁群算法的基本原理.....	12
4.1.2 传统蚁群算法的改进.....	14
4.1.3 改进蚁群算法具体流程.....	16

4.2 多个小车的路径规划算法设计.....	17
4.2.1 碰撞类型分析.....	17
4.2.2 贪心思想引入.....	18
4.2.3 预约表方法.....	19
4.2.4 单行道规则.....	20
<b>第5章 仿真实验分析.....</b>	<b>22</b>
5.1 仓库环境描述.....	22
5.2 仿真实验结果分析.....	22
5.2.1 任务生成信息.....	23
5.2.2 碰撞分析.....	23
5.2.3 任务完成状态分析.....	26
<b>第6章 总结与展望.....</b>	<b>27</b>
6.1 总结.....	27
6.2 展望.....	28
<b>参考文献.....</b>	<b>29</b>
<b>致谢.....</b>	<b>31</b>

## 第 1 章 前言

### 1.1 研究背景与研究意义

现如今,随着经济、科技水平的不断发展,人们的生活水平出现日新月异的提高,其显著表现之一是物流行业的迅猛发展。每年双十一、双十二等节日,越来越多的人群会选择网上购物的渠道进行消费;并且,网上购物的门槛随着智能手机的推广而不断降低,即便是老年人也逐渐接受网上消费的方式。根据京东公布数据显示,2021 年截至 11 月 11 日 23 时 59 分,双十一累计下单金额超 3491 亿元。

但是,巨额下单金额背后也存在着许多问题。其中之一便是,在双十一等消费需求骤增的时间节点,传统物流行业面临着巨大的货品压力。要在短时间内实现对巨大数额货物的入库分拣、储存以及出库,仅仅依靠人力便显得十分困难。因此,设计出一个足够“智能”的仓库,成为物流仓储研究的重中之重。其“智能”指,仓库可以脱离人工,依靠设计好的程序实现一系列从货品入库到货品出库的操作。

这一点的重要性,不仅在于物流货品量大的场景(如双十一、双十二等节日),而且,在平时仓储作业中,“智能”仓库的参与可以极大地解放物流行业对于劳动力的需求,减少仓库运营成本。传统物流行业是典型的劳动密集型产业,具有需要大量人力、效率低、易出错、人工成本高的特点。如今,中国面临着严重的人口老龄化问题,劳动力的缺口逐渐产生,这对许多依赖廉价劳动力的传统行业势必会带来一定的冲击。因此,我国传统物流行业由“重”至“轻”的转型革新已然迫在眉睫。“智能”仓库也能够实现许多仅靠人力无法实现的功能,如:统筹管理货品的入、出库策略,动态监管仓库内的货物情况等,这些功能能够一定程度上提高仓库的运作效率。

综上所述,智能仓库的智能化管理已成为物流行业研究的重点之一。而智能化管理又包含了以下内容:订单分配、拣选路径规划、装箱等。脱离人工的智能分拣是保证物流过程中对海量货物做出正确决策的关键环节之一,而智能机器人的路径规划<sup>[1]</sup>则是实现智能分拣的重要保证。本文主要针对拣选路径规划问题,总结归纳前人对于单个机器人路径寻优的研究内容;并且,尝试从分析多机器人碰撞原因的角度,参考



贪心思想，将传统蚁群算法推广并应用到多机器人避撞问题。最后，针对多机器人仓库和存在电量损耗的仓储环境进行仿真模拟。

## 1.2 路径规划的任务描述以及算法现状

现如今，国内外的知名电商均已率先实现了智能仓储。通过智能仓储，实验表明智能仓的拣货效率远超人工传统仓的拣货效率极限，并且这尚不是智能仓的拣货效率极限。

本文主要针对路径规划问题进行探讨研究。路径规划是机器人智能分拣领域的重要组成部分，其旨在为有障碍物的环境，搜索并规划出一条从起点到终点、距离最短的无碰路径。其优劣评价标准由：最优路径长度、转角大小以及转弯次数<sup>[2]</sup>等。目前，国内外在这一方面的诸多研究已取得显著成果，如A\*算法<sup>[3]</sup>、人工势场法<sup>[4]</sup>、粒子群算法<sup>[5]</sup>、遗传算法<sup>[6]</sup>、蚁群算法<sup>[7]</sup>、深度强化学习算法<sup>[8]</sup>等。

蚁群算法是一种启发式仿生学算法，在处理路径寻优问题上有比较好的应用效果，也被广泛投入研究。因此，本文主要针对蚁群算法的相关优化方案展开模拟分析。蚁群算法的主要问题集中在：如何尽量跳出局部最优，实现全局最优并保证算法收敛的效率。针对这一关键问题，不少学者提出了相关的优化方案。文献<sup>[9]</sup>中，Hu Jiabin 等针对蚁群算法收敛速度慢，规划最优路径存在冗余拐点从而不是全局最优的问题，提出优化的多步长蚁群算法，通过增加移动机器人的视野实线路径的平滑处理，并且通过差异化更新信息素提高收敛算法收敛速度，但同时对于当前蚂蚁可行节点过多会增加算法的随机性，不利于算法的收敛。文献<sup>[10]</sup>中，陈晖等针对蚁群算法初期寻优效率慢、收敛速度慢的问题，提出加入先验地图信息，非均匀初始化不同地图区域信息素浓度来避免算法初期蚂蚁盲目寻找路径；并且通过引入奖惩机制，动态调整信息素挥发系数，提高算法收敛速度，但奖惩机制会降低解的多样性，使得算法更容易陷入局部最优。文献<sup>[11]</sup>中，刘建华等将人工势场算法与传统蚁群算法相结合，通过势场力引导信息素扩散来避免蚁群盲目搜索，提高了蚁群对障碍物的预避障能力。文献<sup>[12]</sup>中，张强等以改进的人工势场算法优化蚁群初始路径筛选，缓解初始路径交叉影响收敛速度问题，并且设计负反馈通道构建信息素自适应更新，保证算法在提升收敛速度的同

时保证全局搜索能力。

### 1.3 论文主要工作

本文主要侧重对于前人研究成果的学习与继承，秉承蚁群算法易于其他算法相结合的优点，通过吸取诸如A\*算法的设计思路来对传统蚁群算法进行优化改进，并且将改进后的蚁群算法应用于智能仓储路径寻优问题中。除此以外，传统蚁群算法只能解决单个移动机器人的路径规划问题，本文引入对多个机器人碰撞机制的分析，结合贪心算法，设计出一套基于改进蚁群算法的避撞系统。同时，建立起一套智能仓库的调度可视化系统，对多个移动机器人进行任务分配以及路径规划，实现高效且无碰撞地实现入库、出库任务。模拟结果通过 Python 给出，证明算法的可靠性。

### 1.4 论文主要架构

第1章为前言部分，主要介绍本论文的研究内容、背景与研究的现实意义，相关算法的进展以及论文的主要工作；

第2章为环境建模部分，主要介绍本论文基于“膨胀法”与“栅格法”，根据现实仓库环境构造出可操作可实现的数学模型，并对移动机器人在该数学模型中的作业情况和相关假设条件进行说明；

第3章为路径规划任务分析部分，我们指出单个移动机器人路径寻优是P问题、多个移动机器人是NP-HARD问题。对于后者，现在尚不存在一个多项式时间内求解的算法（但是这个问题的解是可以在多项式时间内被验证的）。基于对问题的可解性以及原问题的忠诚性考量，我们采用一些启发式算法去近似地处理该问题，本文主要考虑蚁群算法及其改进；

第4章通过将蚁群算法与A\*算法相结合，解决单个机器人的路径规划；通过贪心思想将复杂问题分解成简单子问题，并且分析碰撞类型，引入预约表机制并加以改进，从而解决多个机器人路径规划中的碰撞问题；

第 5 章为仿真实验，给出一个特定的仓库模型进行模拟。通过格点搜索的思想得到蚁群算法的最佳参数。利用该组参数，对整个系统从分配任务至各个机器人协同完成任务的全过程进行仿真实验；

第 6 章为总结与展望，总结本文的研究成果并对相关推广角度展开细节讨论。

## 第2章 智能仓库建模

### 2.1 智能仓库系统运作流程描述

为了满足人们对物流需求的不断增长，建设智能仓库系统来代替人工拣选、管理货物已成为现代物流行业提高工作效率的必经之路。其效率主要体现在，在相同时间内，分拣、储存货品的数量以及正确率。本文考虑到，对于“单车单任务”的模拟实验效果更加直观，因此在后续的模拟仿真中，依旧沿用这一作业模式。但是，这里指出，本文设计的对于多移动机器人避障的算法完全可适用于“单车多任务”的作业模式，能够很好的适应实际仓库场景中多样化的作业模式。

智能仓库强调多设备协同工作，通过一系列环节互相配合，从而实现货品入库、分拣、上货、出库等功能高效、安全地完成。本文主要侧重于多移动机器人无碰撞路径规划。这一问题贯穿于整个智能仓库的调度任务中，是极为重要的一个环节。对于其他设备，本文均默认为已经设定好的系统进行模拟仿真。

在本文构建的智能仓库系统中，主要包括以下几个设备：入库拣选台、出库拣选台、充电桩、附带堆垛机的货架以及移动机器人。下面给出仓库运作的具体流程：

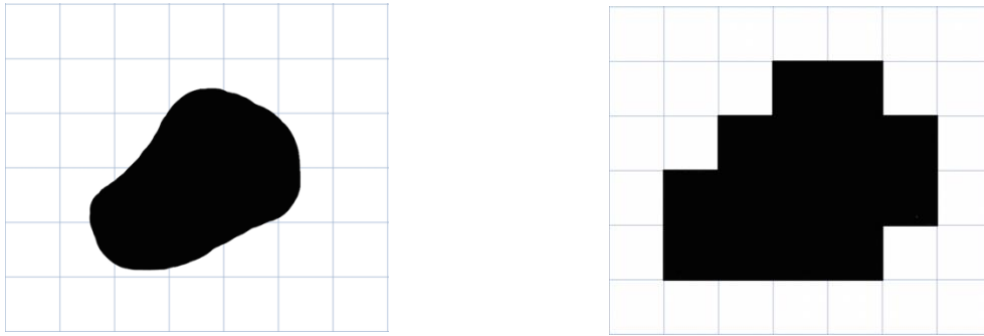
1. 生成订单，将该订单分解成一系列任务。任务内容包括：拣选台位置（入库或出库）、货架位置、货品数；
2. 将分解产生的一系列任务分配给空闲且电量充足的移动机器人；
3. 移动机器人收到任务后，得到对应任务的最佳移动路径，从而完成相应任务；
4. 系统对完成任务的移动机器人进行检查，是否产生电量不足情况。若是，返回充电桩处充电；若否，标记为空闲状态等待下一个任务；
5. 对于状态空闲的小车，系统实现任务的再分配。若全部任务均已完成，则设置所有移动机器人均返回充电桩处进行充电待命。

## 2.2 基于栅格法的环境建模

在三维的现实仓库环境中，移动机器人的路径移动发生在地面上。因此，移动机器人的工作环境可抽象成二维平面，所有的实验模拟均发生在这一二维平面上。

栅格法是移动机器人环境二维建模中一个常用的方法，其基本思想为：将移动机器人连续动作状态离散化，将移动空间离散成为若干的格点。其中，这些格点将根据可通过性的不同分为障碍格点和自由格点。这样处理的好处是：我们可以借此缩减移动机器人在某一位置的后续可行方向至有限个，便于我们路径规划的实现。

在障碍格点的建模问题上，基于安全性的考量，我们采取“膨胀法”进行处理。“膨胀法”的具体操作流程如下图所示：



(1) 原始障碍

(2) 膨胀法

图 1. 障碍物的栅格化

通过“膨胀法”，我们将原始障碍的边界轮廓膨胀至充满一整个单元格。我们规定左下角为坐标原点，即设置 $(x, y) = (0, 0)$ 。我们通过一个示性函数 $\gamma(x, y)$ 来用数学语言描述对于仓库环境的建模过程：

$$\gamma(x, y) = \begin{cases} 0 & \text{(自由栅格)} \\ 1 & \text{(障碍物栅格)} \end{cases}$$

这里，决定栅格法环境建模效果好坏的关键因素之一在于：如何设置单个栅格的大小。过小的栅格会导致计算复杂度大大增加，影响整体路径规划的效率；过大的栅格会导致环境建模过于粗糙，无法很好地还原原始环境中的所有信息。

为了便于我们对路径寻优问题的求解，我们对仓库环境做出以下必要假设：

1. 在同一时间内，同一个栅格内只能容纳一个移动机器人；

2. 对于移动机器人的路径规划而言，三维智能仓库的信息可完全被二维仓库平面所决定。换言之，我们不考虑智能仓库高度方向对于移动机器人路径规划的影响；
3. 障碍物位置、分拣台位置以及上下货货架位置均为已知信息；
4. 不考虑意外情况对智能仓库环境的影响，如：货物从货架上意外掉落等情形。

## 2.3 移动机器人运动建模

移动机器人是智能仓库的调度任务中，实现智能仓库各项功能（入库、出库等）的载体。其在传统物流行业的普及，可以充分减少企业对于劳动力的投入以及工人的重体力工作，将更多的投入分配到系统的监管与控制中。因此，我们需要保证移动机器人在工作过程中，可以始终保持着高效、稳定、安全的工作状态。因此，我们必须建立起移动机器人严谨的运动模型。

本文主要考虑通过制定一个运动规则（如四方向、八方向、单行道等），从全局上控制移动机器人在系统中的运动机制。我们将所有在 $(x,y)$ 点处理论可行点我们记为 $Policy(x,y)$ 。这里， $Policy(\cdot, \cdot)$ 定义为一个集合映射：

$$Policy: \mathbb{Z}^2 \rightarrow S$$

其中， $S$ 表示仓库中所有位置构成的集合，而四方向、八方向运动规则则如下图所示：

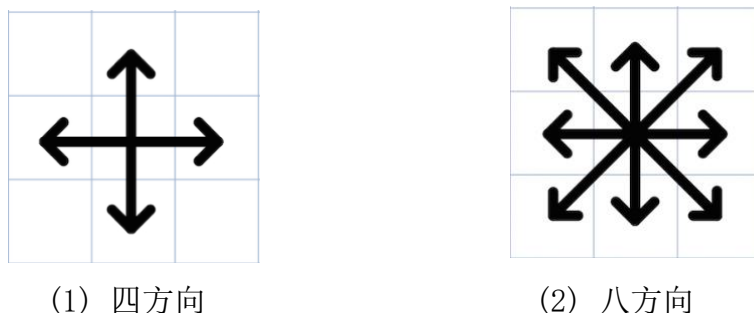


图 2. 具体运动规则示例

如图 2 所示，如果我们指定运动规则为四方向运动，那么在位置 $(2,3)$ 处对应的理论可行点  $Policy(2,3) = \{(2,2), (2,4), (3,3), (1,3)\}$ 。

根据 $Policy(\cdot, \cdot)$ ，我们可通过栅格法示性函数 $\gamma(x,y)$ ，实现对移动机器人处于位

置 $(x, y)$ 时，所有可行后续移动格点 $F(x, y)$ 进行严格数学刻画：

$$(a, b) \in F(x, y) \Leftrightarrow \begin{cases} (a, b) \in Policy(x, y) \\ \gamma(a, b) = 0 \end{cases}$$

这里， $F(x, y)$ 指坐标 $(x, y)$ 处的所有后续可行移动格点集合； $Policy(x, y)$ 指一个特定移动规则下的所有理论可行点（如四方向、八方向规则）。在这种栅格法示性函数 $\gamma(x, y)$ 的帮助下，我们可以数学化对于移动机器人可行方向的定义。

为了便于对路径寻优问题的求解，我们对小车运动模式做出以下必要假设：

1. 假设小车严格按照指定的 $Policy(\cdot, \cdot)$ 进行移动，不会发生错误移动情况；
2. 假设小车当需要沿斜线移动时，速度提升，从而使得完成一次斜线运动的时间与完成一次直线运动的时间严格相等；
3. 假设小车所有运动均为匀速直线运动，不考虑加速度的影响；
4. 假设小车运动过程中均沿着中心行驶；
5. 假设小车运送货品的重量对小车整体运动状态不产生影响。

## 2.4 智能仓库具体模型

根据以上讨论，我们设计仿真模拟的智能仓库平面图，如下图所示：

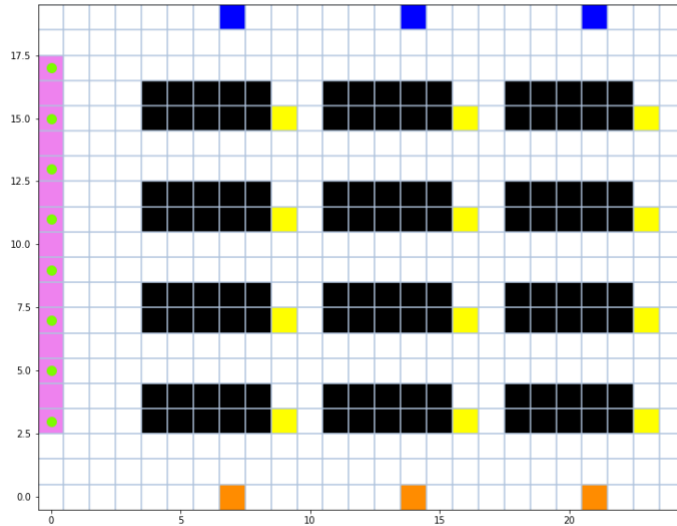


图 3. 仿真模拟的智能仓库平面图

图 3 中通过不同颜色的区域来区别并表示智能仓库中的不同设备：黑色区域表示障碍物栅格，其余颜色区域表示自由栅格；橙色区域表示入库拣选台；黄色区域

表示附带堆垛机的货架；蓝色区域表示出库拣选台；粉色区域表示充电桩。

并且，我们也对移动机器人采用不同颜色来代表其不同的工作状态：绿色表示无货物状态；黑色表示移动机器人上载有货物状态；红色表示电量不足状态；黄色表示充电电量至 50%状态。



## 第 3 章 路径规划任务分析

### 3.1 单小车路径规划问题分析

首先，我们先对单个小车的路径规划问题进行分析。这里指出，在离散前提下，单个小车路径规划问题是一个 P 问题（存在多项式时间内求解的算法）。

**证明：**单个小车的路径规划问题，即给定起点、终点以及障碍物点，找到一条从起点到终点并且不经过障碍物的道路。因此，根据问题描述，我们可以将其抽象为：在一个图 $G$ 上，从起点到终点的最短路问题，其中，可行方向根据障碍物约束给出。显然，我们可以通一个多项式算法，根据原问题建立起对应的图 $G$ 。并且，对于一个图上的最短路问题，我们是存在多项式时间内求解的算法（比如动态规划算法，时间复杂度为 $O(N^2T)$ ）。因此，我们将原路径规划问题通过多项式算法转化为一个 P 问题（图上的最短路问题）。故单个小车的路径规划问题是 P 问题。

因此，对于单个小车的路径规划问题，我们可以找到多项式时间内求解的算法。但是，对于 P 问题，我们依然可以选择一些启发式算法，如：蚁群算法、遗传算法等。这些算法的好处是，我们的计算成本会下降，同时，解的质量依然有一定程度的保障；其不足是，我们无法保证我们得到的解一定是全局最优解（但这是一个比较好的局部最优解）。

### 3.2 多小车路径规划问题（Multi-Agent Pathing Finding）分析

多个小车的路径规划，即指为多个智能体寻找出从起点到终点无碰撞的道路。这里，我们指出，多个小车的路径规划问题属于复杂的组合优化问题。该问题的状态空间随着小车（智能体）数量的增加成指数级增加，现已被证明属于 NP-HARD 问题。因此，对于这一大类问题，现今不存在任何多项式时间内的求解算法。基于这一点，我们考虑放宽我们的要求，我们不要求寻找到全局最优解，而考虑得到一个高质量的解（局部最优解）。

因此，我们建立去对于此问题的问题分析：对于多个小车无碰撞路径规划问题的求解，我们采取贪心思想、改进蚁群算法（与 $A^*$ 算法相结合）和预约表方法相结合的处理思路。首先，将改进蚁群算法作为处理单个小车路径规划问题的基算法，在保证解的质量的前提下减小计算成本；其次，基于贪心思想，我们考虑根据任务分配先后顺序，依次对接受任务的小车进行路径规划，保证其与所有正在执行的小车均无碰撞发生；最后，我们通过预约表方法来具体实现无碰撞的路径分配，在为小车分配路线时，先将系统中所有正在工作的小车路径加入预约表，再对该小车进行路径规划，从而保证与其他已有任务的小车均无碰撞发生。

## 第 4 章 小车路径规划算法设计

### 4.1 单个小车的路径规划算法设计

这里我们的基算法采用蚁群算法，蚁群算法作为全局路径搜索算法，具有原理简单、易于其他算法相结合的优点。我们将传统蚁群算法作为我们算法设计的基础，利用A\*算法与其相结合，根据A\*算法的原理去设计传统蚁群算法的参数更新结构，并且通过一些其他的算法去调整蚁群算法中参数的设置。具体算法细节如下：

#### 4.1.1 传统蚁群算法的基本原理

蚁群算法最早由 DORIGO 于 1996 年提出，最早应用于解决 TSP 问题上（NP-HARD 问题）。蚁群算法的原理是，通过模拟蚂蚁寻找食物的过程，在寻路过程中与环境产生交互，根据信息素的正反馈机制来实现最优路径的寻找。

蚁群算法的关键步骤有两个：1. 概率选择机制；2. 信息素更新机制。前者是指了蚂蚁当面临岔路时，会根据先前蚂蚁留下的信息素选择出一条较优的路；后者是指先前的蚂蚁会在经过道路时，会留下一一种叫做信息素的化学物质。这种物质会随着时间流逝而不断挥发，也会随着走过的蚂蚁数量的增加而不断增强。

关于概率选择机制，传统蚁群算法试图通过 $[0, 1]$ 区间内均匀分布随机数来模拟这一过程。蚁群算法主要通过信息素浓度以及启发信息来对未来下一步可能格点分配一个概率常数，其路径选择规则如下：

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha (\eta_{ij}(t))^\beta}{\sum_{j \in a_i} (\tau_{ij}(t))^\alpha (\eta_{ij}(t))^\beta} & (j \in a_i) \\ 0 & (j \notin a_i) \end{cases}$$

其中，各个记号的意义为：

$p_{ij}^k(t)$ ：表示时刻 $t$ ，蚂蚁 $k$ 从状态 $i$ 转移到状态 $j$ 的概率；

$\tau_{ij}(t)$ : 表示时刻 $t$ , 身处状态 $i$ 时, 状态 $j$ 的信息素浓度;

$\eta_{ij}(t)$ : 表示时刻 $t$ , 身处状态 $i$ 时, 状态 $j$ 的启发信息浓度;

$\alpha$ : 表示信息素浓度因子;

$\beta$ : 表示启发信息浓度因子;

$a_i$ : 表示处于状态 $i$ 时, 可转移格点的集合（先前用记号 $F(i)$ 来表示, 此处为了和文献内记号一致, 选用记号 $a_i$ ）。

对于传统蚁群算法, 一般如下选择启发信息浓度:

$$\eta_{ij}(t) = \frac{1}{d_{jg}}$$

其中,  $d_{jg}$ 表示状态 $j$ 到终点（状态 $g$ ）的欧式距离。

此信息素更新机制主要包括两部分: 1. 上一时间节点信息素的挥发; 2. 该时间节点上因蚂蚁经过该节点而导致信息素增加。前者被称为全局信息素的更新, 相应地, 后者被称为局部信息素的更新。该信息素的更新机制, 兼顾全局搜索能力和局部搜索效率。但值得指出, 过强的局部信息素的更新, 即有助于正向引导后续蚂蚁对于最优路径的寻找, 但也会使得整体很容易陷入局部最优。相应的公式为:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}^{best}(t, t+1) \quad \forall (i, j) \in T_{best}$$

$$\Delta\tau_{ij}^{best}(t, t+1) = \frac{Q}{L_{best}}$$

其中, 各个记号的意义为:

$\rho$ : 信息素挥发因子 ( $\rho < 1$ );

$\Delta\tau_{ij}^{best}(t, t+1)$ : 时间 $t$ 至时间 $t+1$ , 考虑“精英”蚂蚁带来的信息素增益;

$T_{best}$ : “精英”蚂蚁的对应路径;

$L_{best}$ : “精英”蚂蚁的最优路径长度。

直觉地, 这种信息素更新策略主要强调两方面: 1. 通过信息素的挥发来削弱当前时刻信息素对下一时刻信息素的影响, 从而引导蚂蚁逃离局部最优; 2. 通过“精英”蚂蚁信息素增益来引导蚂蚁整体普遍更倾向于搜索好的路径, 从而促进向优秀路径收敛。综合这两方面可以看出, 传统蚁群算法的设计能够同时保证全局收敛性与收敛速度。

## 4.1.2 传统蚁群算法的改进

这里，对于传统蚁群算法的改进，这里主要分成以下三个方面：

1. 基于强化学习在无模型学习（Model-Free）领域的 $\epsilon - greedy$ 状态选择策略（探索与利用）对传统蚁群算法概率选择机制进行改进；
2. 基于 $A^*$ 算法以及启发信息诱导因子对传统蚁群算法启发信息因子进行改进；
3. 基于批更新（Batch-update），对传统蚁群算法信息素更新机制进行改进。

### 1. 对于算法概率选择机制的改进

我们实际一个概率阈值 $\epsilon_0$ ，以概率 $\epsilon_0$ 进行探索（按照传统蚁群算法进行状态选择），以概率 $1 - \epsilon_0$ 进行利用（采用贪心策略，选取转移概率最大的可行节点作为下一个状态）。即，改进的概率选择机制为：

$$j(i, t, k) = \begin{cases} \operatorname{argmax}_{j' \in a_i} p_{ij'}^k(t) & U \geq \epsilon_0 \\ J & U < \epsilon_0 \end{cases}$$

其中，各个记号的意义为：

$j(i, t, k)$ ：时间 $t$ 时，第 $k$ 只处于状态 $i$ 蚂蚁在 $t + 1$ 时的状态；

$J$ ：概率分布函数为 $\{p_{ij}^k(t)\}$ （固定 $i, k, t$ ）的随机变量。

此改进可减轻蚁群算法蚂蚁盲目搜索的现象，增强算法的全局收敛性；并且保留原有概率选择机制，保证算法结果的全局最优性。

### 2. 对于启发信息因子的改进

传统 $A^*$ 算法是一种广度优先的路径搜索算法，主要将从当前状态经过状态 $i$ 最后到目标状态的距离估计值 $f(i)$ 分解成两部分：第一部分是当前状态到状态 $i$ 的距离函数；第二部分是状态 $i$ 到目标状态的启发式距离函数，一般选择为曼哈顿距离。即：

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

除此以外，考虑到在搜索前期，信息素分布较为均匀，蚂蚁倾向于盲目移动这一现象，我们设计启发信息诱导因子 $\xi \in (0, 0.5)$ 。加入启发信息诱导因子，可以强化启发信息在搜索前期的作用，避免初期浪费过多计算性能在蚂蚁盲目搜索中；并且，为了避免在算法后期因为过度强化启发信息而使得最优解陷入局部最优（停滞更新），我们弱化启发信息在迭代后期的作用。

基于这两点考量，我们考虑对改进的到如下自适应更新启发信息：

$$\eta_{ij}(t) = \frac{1-\xi}{\xi} \frac{N_{max} - N_{cur}}{N_{max}} \frac{1}{f_i(j)}$$

其中，各个记号的意义为：

$\eta_{ij}(t)$ ：在时间 $t$ （ $N_{cur}$ ）处，身处状态 $i$ 时，状态 $j$ 的启发信息浓度；

$\xi$ ：启发信息诱导因子（ $0 < \xi < 0.5$ ）；

$N_{max}$ ：最大迭代次数；

$f_i(j)$ ：基于 $A^*$ 算法的从当前状态 $i$ 经过状态 $j$ 最后到达目标状态的距离估计值。

该改进后的启发信息可以实现随着迭代次数增加的自适应调整：当迭代次数较少，即算法初期，对启发信息浓度进行增强（ $\frac{1-\xi}{\xi} > 1$ ， $\frac{N_{max}-N_{cur}}{N_{max}} \approx 1$ ）；当迭代次数比较大，即算法末期（ $N_{max} \approx N_{cur}$ ），对启发信息浓度进行抑制（ $\frac{N_{max}-N_{cur}}{N_{max}} \approx 0$ ）。这样生成的启发信息更能够保证算法初期的寻路效率，从而加速算法的收敛速度；也能够保证在算法末期，整个蚁群系统仍能够有较强的随机性，保留逃脱局部最优、实现全局最优的可能。

并且，由于小车运行环节为有障碍物的仓库，因此通过欧式距离作为衡量节点状态至目标状态的距离估计是不合理的。因此，我们基于 $A^*$ 算法的启发式距离（曼哈顿距离）对传统蚁群算法中的欧式距离 $d_{ij}$ 进行修正，从而保证在智能仓库寻路的任务场景下，启发式因子 $\eta \propto \frac{1}{distance}$ 中距离项的合理性。

### 3. 对于信息素更新机制的改进

传统的蚁群算法对信息素更新一般采用仅对最优路径经过的状态节点进行信息素的增强，这样的更新策略只将最好的路径信息通过信息素的方式正向反馈给后续时间点的状态选择。因此，这种反馈策略很有可能会忽略其他一些次最优路径所包含的对全局最优有帮助的路径信息，从而造成路径更新效率降低，受困于局部最优解。

基于此考量，我们提出路径长度中位数作为评价路径好坏的基准。将路径长度小于（优于）中位数的路径经过的各个状态节点均进行信息素的增强，即：

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \sum_{T_k \in T_{better}} \Delta \tau_{ij}^{T_k}(t, t+1) \quad \forall (i, j) \in T_k \in T_{better}$$

$$\Delta\tau_{ij}^{T_k}(t, t+1) = \frac{1}{|T_{better}|} \frac{Q}{L_k}$$

这里， $T_{better}$ 即为上文提到的路径长度小于中位数的路径集合； $T_k$ 是其中第 $k$ 条优秀路径； $L_k$ 是其对应路径的长度；其与记号均与传统蚁群算法一致。

相比于传统的信息素更新机制，改进后的更新机制由于引入批更新，会使更新曲线更加柔滑。并且，更新后的算法将能够获得更多优秀路径的节点信息，而不是单单最优路径的节点信息，这将有效会提高我们的全局路径寻优算法的全局性。

### 4.1.3 改进蚁群算法具体流程

图 4 为改进蚁群算法流程图：

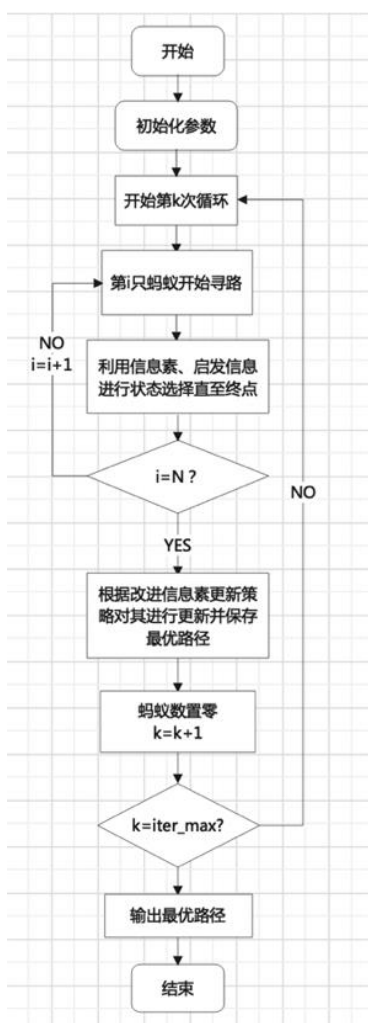


图 4. 改进蚁群算法流程图

具体步骤为：

1. 开始阶段，初始化信息素矩阵 $\tau_{ij}(0)$ ，信息素启发 $\alpha$ ，启发信息因子 $\beta$ ，启发信息诱导因子 $\xi$ ，信息素挥发因子 $\rho$ ；输入初始位置、结束位置，最大迭代次数 $K_{max}$ ，蚁群数量 $N$ ；
2. 设置迭代器 $k = 1$ （第一次循环）；
3. 设置蚂蚁计数器 $n = 1$ （第一只蚂蚁），每一只蚂蚁完成一次路径探索后调整计数器 $n = n + 1$ ，至 $n = N$ 停止。在每次寻路后，若成功寻找到终点，保存路径及长度；若失败（陷入死锁），标记并释放下一只蚂蚁；
4. 当 $n = N$ 时，通过记录的路径长度以及路径节点更新信息素矩阵，重置计数器至 $n = 1$ ，并调整迭代器 $k = k + 1$ ，重复操作(2)至 $k = K_{max}$ 停止；
5. 当 $k = K_{max}$ 时，输出最优路径，改进蚁群算法终止。

## 4.2 多个小车的路径规划算法设计

我们考虑基于对单个小车的路径规划算法去解决多个小车无碰路径规划问题。其主要困难有如下两点：

1. 障碍物会随着系统中正在运行的小车而处于动态更新状态，而传统蚁群算法只能够适用于静态障碍物的场景；
2. 由于小车数量增加，整个系统的状态数目呈指数级增加，致使整个问题成为NP-HARD问题。为了解决这两个困难，我们引入贪心思想以及预约表方法来找出一组（次）最优的无碰撞道路。

### 4.2.1 碰撞类型分析

首先，我们进行多小车路径规划中可能会遇到的碰撞类型分析：



### 1. 相向碰撞



图 5. 相向碰撞

如图 5 所示，两辆小车相向而行的两车会因按照既定方向行驶而产生碰撞；

### 2. 节点堵塞碰撞

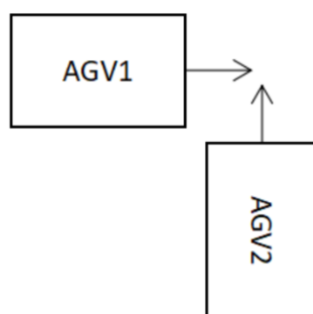


图 6. 节点堵塞碰撞

如图 6 所示，两辆按照既定方向行驶的小车会因在同一时间点占用相同的状态格点而产生节点堵塞碰撞。

## 4.2.2 贪心思想引入

由于多小车无碰路径的规划问题是 NP-HARD 问题，因此为每个小车从全局最优角度寻找一条无碰撞的道路（考虑每个小车在每个时间点的位置均为动态障碍）很难在可接受的时间内给出一个好的结果。因此，我们考虑通过贪心思想简化原问题来权衡计算效率与解的质量的利弊，具体如下：我们按照任务分配顺序进行优先（贪心）路径选择，先分配到任务的小车优先进行其路径选择，这些路径对于后分配到任务的小车均是固定的。换言之，在后分配到任务的小车视角下，这些路径被视为障碍，后分配到任务的小车尝试去寻找路径来避开先分配到任务的小车的路径。

在原问题中，每个小车均可改变自己的路径来使整个系统无碰撞，这固然会使得到的路径效率更加高，但其中带来的计算成本的增加我们往往无法承受。在基于贪心算法对原问题的简化版本中，我们只需要按照任务分配顺序来对当前小车进行路径规

划，固定系统中正在运行的其余小车的路径。通过引入贪心算法，我们实际上把一个复杂的问题分解成多个相对简单的子问题（单个小车动态障碍路径规划），大大缩减了问题的状态空间，增加了问题的可解性；并且，我们针对子问题的求解算法是相当高效且效果好的，这也能够给我们解的质量带来很大程度上的保证。

### 4.2.3 预约表方法

通过贪心算法，我们将复杂的多小车无碰路径规划问题转化为了相对简单的单个小车动态障碍路径规划问题。为了解决这一子问题，我们采用预约表方法来处理节点堵塞碰撞情形。我们将其余小车的运动轨迹生成为一张预约表进行储存，我们的目标转化为：为当前小车找到一条与这张预约表无交集的路径。

通过分析，我们再次细化节点堵塞碰撞问题，并分成以下两大类问题：

#### 1. 中间节点堵塞

我们认为这一情形属于是比较容易处理的，我们采取如下措施：

首先，我们不考虑额外的障碍，根据起点与终点生成一条候选路径，如果这条路径不与预约表发生冲突，那么我们即可采用此路径作为该小车的分配路径；

其次，如若产生冲突，且冲突节点并不位于起点、终点（中间节点堵塞），那么我们可以采取将冲突节点通过额外障碍加入预约表，对预约表进行更新，再根据单个小车的路径规划算法（基于更新后的预约表）生成一条新的候选路径；

最后，如果这条候选路径与预约表无冲突，即采用此路径作为小车的分配路径；若有冲突，重复上述分析。

#### 2. 首尾节点堵塞

我们认为这一情形是相对难以处理的，原因如下：1. 我们不能简单将冲突终点加入障碍表中，这样会导致终点位于预约表而造成路径搜索算法无法找到一条从起点到终点路径的问题；2. 我们也不能简单将起点加入预约表中，这样虽然能够找到一条从起点到终点的路径，但是新生成的路径依然会在起点处与预约表产生冲突，从而陷入死循环。因此，对于这种情况，我们采取如下措施：

针对第一点，当我们检测到尾端节点冲突后，我们不选择重新规划路径的算法去

规避碰撞，而是通过在起点处多停留一秒来逃离尾端冲突；

针对第二点，我们考虑建立规则使得系统中无法产生首端节点冲突的碰撞。规则为：在我们为小车规划路径而建立预约表时，我们在预约表尾添加上其余小车在最后一秒所处的位置。最后指出，这一条件是不产生首端节点冲突的充分条件（不必要）。

通过预约表方法并根据这两类碰撞问题进行具体分析，我们可以有效解决第二类（节点冲突）碰撞情形。

#### 4.2.4 单行道规则

至此，我们通过预约表方法及贪心算法，对第二类节点冲突碰撞类型实现了规避。但是，我们认为，凭借预约表方法很难对第一类相向碰撞进行检测，主要原因在于：预约表方法便于检测在同一时间因占用相同状态节点而产生的碰撞。而第一类相向碰撞并不会产生在同一时间两辆车占用同一个状态节点，而是会形成两辆车互相穿过彼此的换位情形。

因此，为了解决这一问题，我们提出了更强的条件，直接对小车在各个状态节点上的所有可行状态（*Policy*）进行修正，从而达到解决第一类碰撞的目的。

这里，我们引入的策略是：单行道规则。即，在同一条直线道路上，我们在前、后两个运动方向中只允许其中之一，换言之，对于任意一个状态节点，其所有可行状态最多只有三个（至少不允许前、后其中之一）。

由此机器人移动策略得到的路径规划，自然而然的避免了第一类节点冲突碰撞类型：在同一条直线上我们只允许同一个方向的运动（或者左、右）。因此，我们不允许两辆车相向运动，规避了第一类相向节点冲突。

值得指出的是，这一移动策略对于路径规划的影响是微乎其微的：可以想像，相比于八方向的移动策略，唯一的不同在于，当我们需要改变运动方向时，我们只需要多花费一个单位的时间进行向左（或向右）移动。但是，多付出这一个单位的时间，可以换来彻底解决所有的碰撞问题，这在碰撞成本高的场景下是十分好的解决方案。并且，采用单行道规则之后，整个仓库系统的运行呈现出十分稳定的状态：目标相近的机器人会沿着类似的轨迹进行移动，这也会给仓库的管理带来便利。因此，对于实

际应用场景来说，这一规则（单行道规则）能够为多机器人无撞路径规划问题提供一个很好的解决方案。

## 第 5 章 仿真实验分析

本文将利用 Python 进行仿真实验的设计。Python，作为面对科学计算，可视化的开源程序，在如今的科研实验中得到越来越高的重视。本仿真实验主要通过每秒串行依次更新每一辆小车的位置来得到小车并行运行的效果，具体设计如下：

### 5.1 仓库环境描述

本文仿真模拟仓库与图三布局一致，即三个入库分拣台，三个出库分拣台，八辆移动小车，十五个充电桩以及十二个附带堆垛机的货架。

在此环境下，本文忽略小车初始加速度，并且规定其在行驶过程中一直处于匀速运动状态，假定其每秒跨越一个栅格。在初始时间处，八辆小车均匀分布于充电桩处，电量均为 100%。当各辆小车收到其任务后，离开充电桩执行任务。在运行过程中，小车每秒损失 1%电，在剩余电量小于 30%返回充电桩充电，充电速度为 3%每秒。当电量补充完毕后，小车接受任务并重新出发执行。当所有任务全部完成后，所有小车最后返回充电桩处进行充电。至此，该仿真实验结束。

不失一般性，本文算例选取单行道规则进行仿真模拟分析。并且，本文针对以第二类节点堵塞碰撞（首尾堵塞碰撞）情形为例展开详细分析。四方向、八方向的仿真模拟分析与其完全一致。

### 5.2 仿真实验结果分析

仿真实验分析可主要分为三部分：

1. 生成任务；
2. 路径规划（无碰撞分析）；
3. 任务完成状态分析。

### 5.2.1 任务生成信息

任务信息表	
(2, 5)	(1, 8)
(6, 12)	(2, 3)
(2, 8)	(2, 2)
(1, 1)	(5, 2)
(3, 10)	(2, 6)

表 1. 任务信息表

在表 1 中，[1, 2, 3]为三个入库分拣台；[4, 5, 6]为三个出库分拣台；其余均为附带堆垛机的货架。通过该二元组，我们即可实现小车任务的确定。

以第一个任务 (2, 5) 为例，其含义为，分配到该任务的小车，需要先规划处从当前位置到二号入库分拣台的路径，再规划出从二号入库分拣台到五号出库分拣台的路径，属于直接的入库-出库任务。

### 5.2.2 碰撞分析

我们着重观察对于第二类节点堵塞碰撞（首尾堵塞碰撞）的仿真模拟分析：

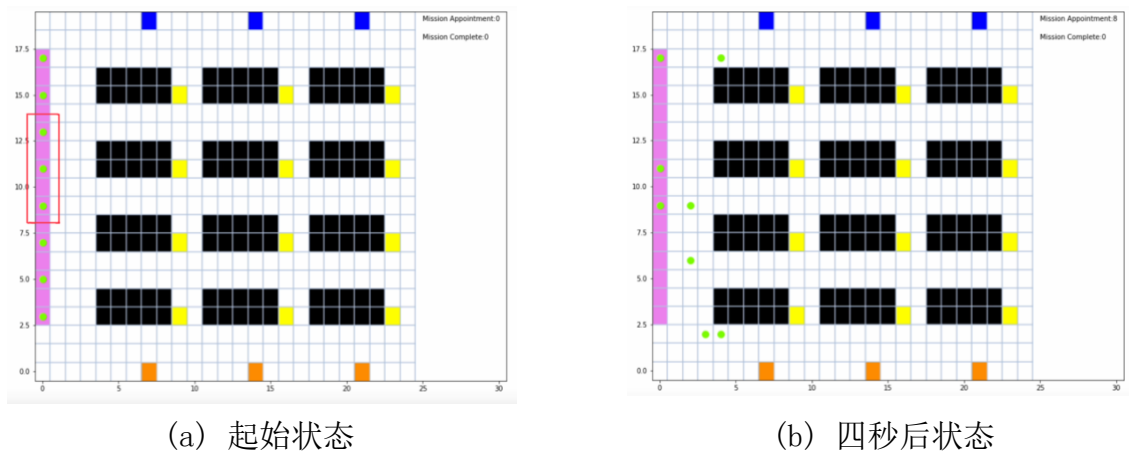
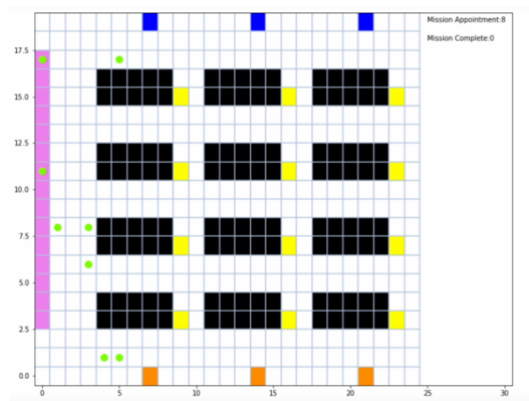
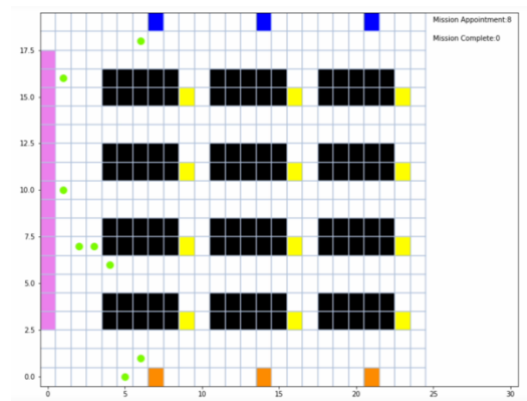


图 7. 首尾碰撞分析 1

我们着重观察图 7(a)中红框内的三辆小车，这里从上至下把所有小车标记为小车 1 至小车 8。在(b)中，我们发现，当小车 3 已经移动三个时间单位后，小车 4 和小车 5 依旧没有移动。至第五、第六秒，小车 4、小车 5 才依次进行移动，如下图所示：



(a) 五秒后状态.



(b) 六秒后状态

图 8. 首尾碰撞分析 2

这一看似反常的现象下，实际上是我们算法底层逻辑的正确反映：由于处于系统的开始阶段，根据鸽笼原理，系统会出现大部分车辆会“争抢”同一个入库分拣台的情况（八辆车、三个入库分拣台）。并且，此时，小车互相的位置是非常靠近的。因此，直觉地，如果相邻的小车的任务出发点相同，那么很有可能他们各自到出发点的最优路径（单个小车路径规划）需要相同的时间。这样，就会造成首尾碰撞发生。这里，这三辆小车恰恰经历了这一过程。

因此，根据我们的避撞机制，小车 3 优先出发，小车 4 在检测到路径冲突后选择原地等待，小车 5 在得到小车 4 的路径后，为了避免与之产生碰撞，在此基础上继续等待一秒。

这里指出，我们选择在起始位置等待（而不是末尾感应到碰撞可能再选择等待）有以下两个好处：

1. 减少了小车在运行-停止等待-再运行导致的对于小车移动装置的损耗以及电量损耗；
2. 能够保证我们分配的路径更加安全（如若在末尾停止，容易造成因设备感应误差而产生碰撞）。并且，如果再系统内运行的车辆过多，选择在即将碰撞前在停止小车、进行等待非常容易产生路径堵塞。我们这样在运行初始阶段进行停车等待的策略

略会大大降低产生路径堵塞问题的可能性。

事实上，系统的后续运行状态可以证明我们上述的分析过程，如图 9 所示：

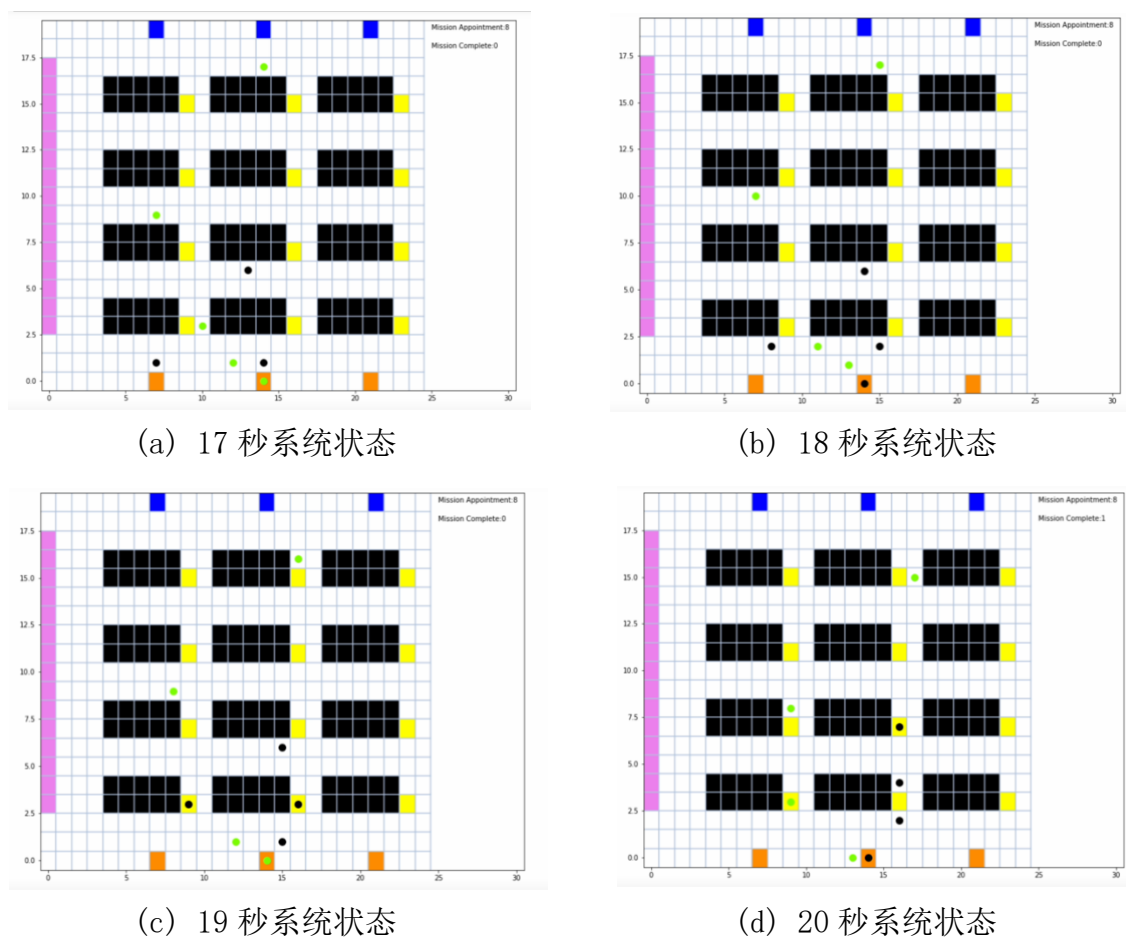


图 9. 首尾碰撞分析 3

可以发现，有四辆小车的任务起点（入库分拣台）均是 2 号入库分拣台。但是，在我们设计的多小车路径避撞算法下，即使面对如此冲突可能性大的任务环境下，各辆小车依旧能够十分协调、有序的运行并完成任务。

此首尾碰撞分析案例可体现出在极端情形下，本算法依然能够很好的为多辆小车规划出合理的路径，避免碰撞的发生。因此，此例能很有力地说明，该多小车路径避撞算法的有效性与可靠性。



### 5.2.3 任务完成状态分析

在任务全部完成后，所有小车均成功返回充电桩位置进行充电：

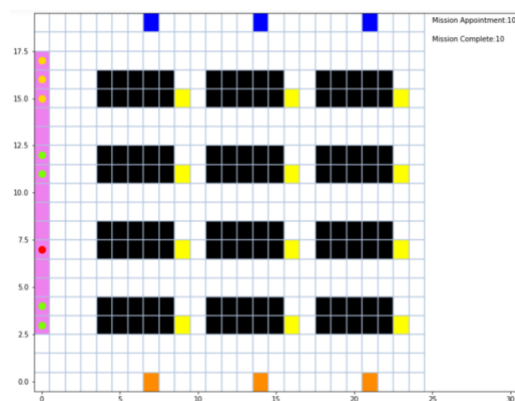


图 10. 所有任务成功完成

至此，该轮所有任务均完美完成，仿真模拟结束。系统开始等待下一轮任务的输入。

## 第 6 章 总结与展望

### 6.1 总结

现如今，随着经济社会的不断发展，人们对优质生活水平的需求也日益增加。正因此，促成了一大批行业的蓬勃发展，传统物流行业正是其中之一。但是，挑战也随之而来。当货品数量没那么多的时候，凭借传统劳动力，尚能满足人们对消费的需要；但是，随着人们的需求飞速增加，仅凭传统劳动力已无法支持这一需求。在此背景下，智能仓库这一概念应运而生。

本文着眼于，在一个普适的智能仓库环境中，切实解决多移动机器人存储货物的无碰路径规划问题。本文的处理思路为：从单机器人路径规划问题入手，选取蚁群算法在计算效率与解的质量中寻找到一个平衡。并且，既根据已有知识储备，又广泛阅读文献、总结前人对于该问题的研究，对该算法在细节上给出自己的优化方案。我们把这一改进后的蚁群算法作为基算法，利用贪心思想、预约表方法以及碰撞分析，最终给出一种处理多移动机器人无碰路径规划的有效算法。我们将复杂问题分解成简单子问题，通过高质量且高效地求解简单子问题，最终实现对于复杂问题的攻克。

最后，陈述本文的主要内容：

1. 利用栅格法、膨胀法进行仓库环境的建模；用数学化语言定义每一个位置状态 $(x, y)$ 上的后续可行位置 $F(x, y)$ ；
2. 利用结合  $A^*$ 算法的改进蚁群算法、贪心算法、预约表方法以及碰撞分析，建立有效解决多移动机器人无碰路径规划问题的具体算法。并且，根据小车不同移动策略，给出两种具体应用方案：八方向移动策略以及单行道策略；
3. 利用 Python 对智能仓库的运行过程进行仿真模拟，证明该算法在解决多移动机器人无碰路径规划问题上的高效性与可靠性。

## 6.2 展望

随着这一年对应用数学领域了解的不断增加，本人深刻意识到对于该智能仓库系统模型的细节还有诸多值得改善的地方：

1. 在该模型中，始终假定小车宏观上处于匀速运动状态中。但是，从微观的角度出发，小车的速度变化必然是连续的。并且，在小车很大一部分的能量损耗会来自于启、停以及拐弯（可以推广至上下货的控制）。因此，自然的，对应这几种状态，我们可以建立起物理模型，根据最小化电量损失，通过微分方程对小车运动状态进行约束，根据最优控制理论寻找该优化问题的最优解。利用此最优控制来在局部控制小车启、停以及拐弯状态。这样从局部优化的角度可以减缓小车电量下降的速度，实现对系统的优化。

2. 本文对解决多小车碰撞的问题主要采用的思想是贪心策略：利用优先级设置先接到任务的小车为高优先级，固定高优先级小车的路径、调整低优先级的小车路径来实现避障。这样的搜索策略能够为无碰撞的好路径提供一个可行解，但是很大可能上会造成局部最优。因此，更加优秀的算法将是真正从算法角度为多小车系统从整体规划出无碰撞的道路，真正从整体最优来考虑问题。从这一角度，我认为可以通过多代理人强化学习理论并结合博弈论（合作）的角度考虑实现。相比启发式算法，这一大类算法具有对收敛性的保证，与神经网络相结合有着非常大的潜力。但是，缺点在于，算法鲁棒性不强，需要对特定环境进行特定处理，相关参数的设计对系统的好坏有着至关重要的作用（奖赏函数的设置等）；得到的最优解依然很可能是局部最优（但有可能会比基于贪心思想产生的解要好）；可能会面临计算量过大的问题。

3. 在任务分配等角度并未深入考虑，任务分配的好坏也会很大程度上决定整个系统的效率。本文未深入分析的主要原因是，我们认为对任务分配可以有一个简单且效果好的处理方案：对订单进行聚类，将尽可能同一类的货物分配给同一辆小车。相比较这个，我们更在意为整个系统寻找到一系列无碰的、高效的路径。这一任务相对任务分配更加不平凡，并且对智能仓库的现实落地起到了至关重要的作用。

## 参考文献

- [1] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(07):961-967. ZHU D Q, YAN M Z. Overview of mobile robot path planning technology [J]. Control and Decision, 2010, 25(07): 961-967.
- [2] Xue Tian, Li Liu, Shuang L, et al. Path planning of mobile robot based on improved ant colony algorithm for logistics[J]. Mathematical Biosciences and Engineering, 2021, 18(4):3034-3045.
- [3] P. E. Hart, N. J. Nilsson, B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [4] Wang D, Chen S, Zhang Y, et al. Path planning of mobile robot in dynamic environment: fuzzy artificial potential field and extensible neural network[J]. Artificial Life and Robotics, 2020, 26 (2) : 129-139.
- [5] SMITH R C, CHEESEMAN P. On the representation and estimation of spatial uncertainty[J]. The International Journal of Robotics Research, 1986, 5( 4) : 56 — 68.
- [6] 魏彤, 龙琛. 基于改进遗传算法的移动机器人路径规划[J]. 北京航空航天大学学报, 2020, 46(04):703-711. WEI T, LONG C. Mobile robot path planning based on improved genetic algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2020, 46(04): 703-711.
- [7] 叶杭璐, 何利力. 基于改进蚁群算法的智慧物流调度规划[J]. 计算机系统应用, 2021, 30(1):7.
- [8] Li B, Liang H. Multi-Robot Path Planning Method Based on Prior Knowledge and Q-learning Algorithms[J]. Journal of Physics: Conference Series, 2020, 1624(4):042008 (9pp).
- [9] 胡佳斌, 王祥澍, 张琪, 等. 优化多步长蚁群算法求解机器人路径规划问题[J]. 传感器与微系统, 2021, 40(10):4.

- [10] 陈晖,周德强.改进蚁群优化算法的移动机器人路径规划 [J].电子测量技术,2020,43(23):17-22. CHEN H, ZHOU D Q. Improved Ant Colony Optimization Algorithm for Mobile Robot Path Planning[J]. Electronic Measurement Technology, 2020, 43(23): 17-22.
- [11]刘建华, 杨建国, 刘华平,等. 基于势场蚁群算法的移动机器人全局路径规划方法[J]. 农业机械学报, 2015, 46(9):10.
- [12]张强, 陈兵奎, 刘小雍,等. 基于改进势场蚁群算法的移动机器人最优路径规划[J]. 农业机械学报, 2019, 50(5):11.

## 致谢

首先，我要感谢本文的指导老师丁睿老师。在大三的数值分析课堂上，我有幸与丁睿老师结识，丁睿老师对课堂抱有无比的热情与耐心，尽心尽力地教授着每一堂课。虽然在当时很遗憾未能和老师有进一步的交流，但是在此次毕业设计中，丁睿老师给予了巨大的帮助，在不断交流过程中让我对所研究的问题的认识不断加深。每当我遇到相关问题时，丁睿老师也会不厌其烦的为我解答相关疑难，给出他专业性的意见，帮我分析之后的改进方向。可以说，毕业论文的顺利完成，完全离不开丁睿老师的尽心尽力的帮助。

其次，我要感谢大学四年里对我们的学习生活无比负责的辅导员周扬老师以及其他在我大学四年里参与教授课程的各位老师。可以说，大学四年我经历了众多风格各异的老师们：有热情似火的马欣荣老师、讲课风趣的王云老师、还有认真负责的丁睿老师等。是他们的辛勤努力，不断丰富着我的知识储备，让我对各个知识领域的理解不断提升。总而言之，非常感谢他们在教职岗位的不断付出，在我的学习道路上给予我莫大的支持与鼓励，让我受益良多。

最后，我要感谢大学四年与我相遇的各位同学。感谢我的舍友以及隔壁宿舍的同学在大学四年对我的照顾。与我同级的张天阳同学，同我在大学四年的生活中互相照料、互相学习；与我同宿舍的贾世闯同学，我们在封校期间克服心中的惰性，坚持学习坚持进步；与我同级的陈政安同学，在学习之处给予我许多建议与帮助，可以说是我今后努力的方向与标杆。

在以后的学习生活中，我会牢记大学所经历的一切，保持热情，继续努力。