浙江工业大学

本科毕业设计（论文、创作）

题目：面向订单分拣的多移动机器人调度算法

作者姓名 汪钰皓

指导教师 赵云波教授

专业班级 自动化 1404

学　　院 信息工程学院

提交日期 2018 年 6 月 23 日

浙江工业大学本科毕业设计论文

面向订单分拣的多移动机器人调度算法研究

作者姓名：汪钰皓

指导教师：赵云波教授

浙江工业大学信息工程学院

2018 年 6 月

Dissertation Submitted to Zhejiang University of Technology for the Degree of Bachelor

Research on Multi-mobile Robots Scheduling Algorithm for Order Sorting

Student: Wang Yuhao

Advisor: Professor Zhao Yunbo

College of Information Engineering

Zhejiang University of Technology

June 2018

浙江工业大学

本科生毕业设计(论文、创作)诚信承诺书

本人慎重承诺和声明：

1. 本人在毕业设计（论文、创作）撰写过程中，严格遵守学校有关规定，恪守学术规范，所呈交的毕业设计（论文、创作）是在指导教师指导下独立完成的；

2. 毕业设计（论文、创作）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，无虚构、篡改试验结果、统计资料、伪造数据和运算程序等情况；

3. 若有违反学术纪律的行为，本人愿意承担一切责任，并接受学校按有关规定给予的处理。

学生（签名）：

年　　月　　日

浙 江 工 业 大 学

本科生毕业设计（论文、创作）任务书

专 业 自动化 班 级 1404 学生姓名/学号 汪钰皓/201403080716

一、设计（论文、创作）题目： 面向订单分拣的多移动机器人调度算法研究

二、主要任务与目标：

1. 了解掌握多机器人调度常规算法；2. 针对订单分拣问题提出新算法；3. 对算法进行分析和实现。

三、主要内容与基本要求：

1. 阅读相关文献，了解该领域基本研究现状；2. 研究多机器人调度相关算法；3. 提出并验证相关的订单分拣算法；4. 撰写毕业论文。

四、计划进度：

2018 开学前 收集相关资料文献，学习相关知识，完成外文翻译、文献综述；熟悉课题，做好开题准备。

第 1-3 周 完成开题报告，参加开题交流

第 4-8 周 提出算法并作验证等工作，接受中期检查

第 9-14 周 进行算法设计和实现等工作。撰写毕业论文初稿

第 15 周 论文修改，毕业答辩，提交相关文档资料

五、主要参考文献：

[1] 《仓储系统中分拣—存取效率优化模型与策略研究》，刘田，华中科技大学，博士论文，2016 [2] 《电子商务配送中心的设计与优化策略研究》，王文蕊，山东大学，博士论文，2014 [3]《配送中心拣货系统优化》，肖际伟，山东大学，博士论文，2010

任务书下发日期 2017 年 12 月 11 日

设计（论文、创作）工作自 2017 年 12 月 11 日 至 2018 年 6 月 12 日

设计（论文、创作）指导教师 赵云波

学科（方向）负责人 杨马英

主管院长 张有兵

面向订单分拣的多移动机器人调度算法研究

摘要

电子商务是一种新兴的基于网络的商业运营模式,买卖双方在网络上确定交易活动,然后通过某种运输方式将所交易的货物运送到买家手中。最近几年，随着互联网以及智能终端的普及，人们越来越青睐于通过网络平台订购商品，既免去了来回奔波的辛苦，又可以享受到较为实惠的价格。近年来，随着电子商务的快速发展，电商之间相互竞争导致物流仓库订单量剧增，随之出现订单延误，“爆仓”等现象，极大影响了用户体验。

为了提升物流仓库处理订单的效率，增加用户体验，同时节约成本，本文针对一种特定格局的仓库，提出利用移动机器人分拣订单的调度算法。并在MATLAB上搭建软件平台，实现了仿真。最后，与一种传统格局的物流仓库订单分拣算法进行了仿真比较，该算法在综合意义上优于传统算法。论文的主要工作如下：

1. 描述了一种物流仓库模型，并且在MATLAB上搭建了仿真软件平台。
2. 根据订单分拣的流程，提出了一种移动机器人的调度算法。
3. 描述了一种传统的物流仓库订单分拣算法，并且将其与本文提出的模型中的算法进行仿真对比。

关键词：电子商务，物流仓库，订单分拣，多移动机器人

RESEARCH ON MULTI-MOBILE ROBOTS SCHEDULING ALGORITHM FOR ORDER SORTING

ABSTRACT

E-commerce is an emerging network-based business operation mode. Both buyers and sellers determine transaction activities on the Internet, and then transport the goods to buyers through some mode of transports. In recent years, with the popularization of the Internet and smart terminals, people have become more and more interested in ordering products through online platforms, which not only eliminates the hard work of going back and forth, but also allows them to enjoy more affordable prices. In recent years, with the rapid development of e-commerce, the competition between e-commerce companies has led to a sharp increase in orders for logistics warehouses, followed by delays in orders, and "explosion of warehouses" and other phenomena, which have greatly affected the user experience.

In order to improve the efficiency of processing orders in logistics warehouses, increase user experience, and decrease costs, this paper proposes a scheduling algorithm for sorting orders using mobile robots for a warehouse with a specific pattern. And set up the software platform in MATLAB, realized the simulation. Finally, compared with a traditional logistics warehouse order sorting algorithm, the algorithm is superior to the traditional algorithm in a comprehensive sense. The main work of the dissertation is as follows:

1. Introduced the logistics warehouse of this specific pattern and the simulation software platform built on MATLAB.

2. The flow of order sorting in the algorithm and the scheduling algorithm of the mobile robot are proposed.

3. Introduced a traditional logistics warehouse order sorting algorithm, and

compared it with the model presented in this paper.

Key Words: E-commerce, logistics warehouse, order sorting, multiple mobile robots

目　　录

第 1 章 绪 论

1.1 课题研究背景及意义

电子商务是一种新兴的基于网络的商业运营模式,买卖双方在网络上确定交易活动,然后通过某种运输方式将所交易的货物运送到买家手中。最近几年，随着互联网以及智能终端的普及，人们越来越青睐于通过网络平台订购商品，既免去了来回奔波的辛苦，又可以享受到较为实惠的价格。

随着 B2C 电子商务的快速发展,为了顺应市场发展趋势，许多知名电商企业如京东和阿里巴巴等纷纷积极筹建自己的物流配送系统，近年来，电商企业之间相互竞争，推出了许多促销活动，导致订单量巨大，而订单配送还必须按时完成，给物流系统带来了不小的压力，比如在“双十一”、“双十二”等促销活动期间,物流仓库经常出现"爆仓"现象,使得用户不能及时拿到自己网购的物品。在这样的情况下,如何提高物流仓库的处理效率,减少"爆仓"现象的产生,成为了物流系统重点关注的问题。物流仓库作业对包括投资和直接操作成本在内的逻辑成本有很大影响。特别是订单分拣被视为最耗费劳动力的操作，因为它通常会耗费 60%-70%的仓库作业成本。换言之，订单分拣主要影响着仓库中的操作性能和成本效率。因此，生成有效的分拣方法和合适的分拣操作对仓库的全局运作效率有着重要影响。

订单分拣是在仓库中的存储位置处取出物品以满足用户要求的一项仓库功能。分拣包括五个步骤：准备，搜索，拣货，运输和其他。在这些步骤中，运输是最重要的，因为它占据了整个过程时间的最大比例（至少 50%），订单分拣次之[1]。当前 B2C 电子商务订单的物流配送呈现出许多新的特点,如客户多、每单需求量小、品类多、存放位置分散,同时顾客对响应时间也有一定的需求。

区别于传统的由人工分拣订单，拣取货物的物流仓库，智能化仓库是一种以人工智能技术为支撑建立的新型自动化仓库，其具有自动化程度高、人力成本低、运转效率高的特点[8]。其中仓储物流机器人在建立智能化仓库的过程中扮演着关键的角色。它能有效降低人力成本，减少出错，提升订单处理效率。目前，以亚马逊的 Kiva 机器人为代表的仓储物流机器人正逐渐开始被应用到智能化仓库建设之中，其在很大程度上可以提高拣选作业的效率。Kiva 机器人作业方式颠覆

了传统电商物流中心人找货、人找货位的模式，而是通过作业计划调动机器人，实现货找人、货位找人，整个物流中心库区无人化，各个库位在 Kiva 机器人驱动下自动排序到作业岗位。数据显示这些机器人每小时可跑 30 英里，准确率达到 99.99%，这种模式的作业效率要比传统的物流作业提升 2-4 倍。[16]

1.2 问题描述

订单分拣问题一般可被分为以下几个子问题[2]：

第一个子问题是订单分批问题（OBP），可以说明如下：给定物品的存储位置、使用的路由策略和拣货设备的容量，如何将客户订单分组到分拣订单中，使得行程的总长度最小化？

第二个是批次分配和定序问题（BASP），客户订单的特点是必须以尽可能最好的方式满足期限，而每个客户订单已经被分配到某个批次中。解决方案由所有订单的总拖延时间进行评估。BASP 可以表述如下：如何将批次分配给有限数量的拣货员，并且对于每个拣货员，应该如何对批次进行排序，从而使总体的迟延最小化？

第三个拣货员路由问题（PRP）可以描述如下：给定一组要从已知存储位置选择的物品，这些位置应该以怎样的顺序被访问，才能使相应的拣货行程的总长度最小化？

1.3 国内外研究现状综述

物流仓库中客户订单指定的物品一般需要在某个特定期限之前提交。违反截止日期可能会延误随后的装载运输过程，结果导致客户满意度大大下降和因此带来的高成本。是否或在多大程度上满足客户订单的截止日期取决于（1）客户订单如何分组到拣配订单（订单分批问题（OBP）），（2）如何将拣配订单分配到订单拣货员（批次分配和定序问题（BASP）），以及（3）每个订单拣货员如何规划路径以收集每个拣配订单的物品（拣货员路由问题（PRP））。这些问题都是密切相关的。因此，目前国内外对于订单分拣的研究主要集中在解决这些问题上。

在 Henn 和 Wäscher(2012)[4]所提的订单分批问题中，他们解决了给定的用户订单该如何结合到订单分拣中使得所有拣货者到达所有需要分拣的物品的必须

路程最短的问题。该作者提出了两种基于禁忌搜索原则的方法来解决这个问题。第一种是经典禁忌搜索（TS），第二种是基于特征的爬山法（Attribute-Based Hill Climber），他们结合节约算法和局部搜索作为基准,研究上面两种算法的有效性。

Henn（2015）[3]把 OBP 与 BASP 结合，导致了联合订单批次，分配和定序问题（JOBASP）。他提出了可变邻域下降（VND）和可变邻域搜索（VNS）的方法来解决 JOBASP。作为上限，他选择了最早的基于开始日期规则（ESDR）的算法，其中客户订单根据其截止日期被分批和排序，并且批次被分配给当前拥有最小总处理时间的拣货员。在其数值实验中包括了多达 200 个客户订单的问题实例。使用基于 ESDR 的算法得到的总延迟平均可以减少 41％（VNS）和 39％（VND）。VNS 算法的应用需要多达 25 分钟的计算时间，而 VND 方法则在最多 30 秒后终止。

AndréScholz 等人（2017）[2]首次将订单分拣的所有子问题一同考虑，因此提出了在有多个拣货员条件下的联合订单分批，批次分配、定序和拣货员路由问题的解决方案。他们基于 Henn 的研究，提出了一种启发式解决方法，即可变邻域下降算法（VND）。他们利用 VND 获得局部最优解后，就使用 Lin-Kernighan-Helsgaun（LKH）启发式方法解决出现的 PRP。该组合方法对于规模比较大的实例依然有较快的处理速度。

在国内的研究中，陈子立等人（2015）[1]提出了 JOBASP 的一种遗传算法，并通过蚁群克隆算法解决了出现的 PRP。然而，在这种遗传算法中，分批和排序问题是分开考虑的，这就导致在考虑到拣货设备的容量限制下有许多不可行的解决方案。此外，蚁群克隆算法消耗的计算时间要远远多于针对特定 PRP 问题的方法。因此，他们只考虑了仅有 8 个订单的问题实例，若处理大型问题，计算时间将是最大的问题。

蔡等人（2008）[5]提出了一种针对 JOBASRP 的遗传算法。在这种算法的第一阶段，需要分拣的订单批次得到确认。第二阶段，这种算法决定这些批次的最短拣货路径，这一方法被用来解决在仓库系统中的复杂批次分拣问题。除了总的迟延之外，他们也将总提前时间和行程总长度减到最小。在他们的方法中，允许客户订单的分割。

刘进平（2010）[6]针对分区和订单品项数量对拣货路径选择的影响和几种启

发式分拣路径在不同订单规模下进行了比较研究，以评价各种拣货路径方法。韦超豪（2016）[7]对面向 B2C 电商的订单分拣优化进行了研究，他将 k-Means 算法和 Canopy 算法结合来处理订单分批问题，首先通过 Canopy 进行初步聚类得到初始的聚类中心点,距离计算方式采取简单的方法(储位相似度)。然后采用 k-Means 算法进行聚类得到最终的分批结果,距离的计算采用两个订单合并之后增加的分拣距离。

1.4 主要研究内容

本文是研究针对特定结构的物流仓库模型提出的面向订单分拣的多移动机器人调度算法，以提高物流仓库订单分拣的效率，提高用户满意度。本文主要结构如下：

第一章是绪论部分，主要介绍了课题的研究背景和意义，对于物流仓库订单分拣模型的国内外研究现状综述，以及本文的主要研究内容。

第二章对研究问题进行了描述和对物流仓库模型进行了一些基本定义。

第三章详细描述了特定结构的物流仓库模型以及在 MATLAB 平台上搭建的软件仿真平台，以及在该平台上实施的订单发放，分配，移动机器人调度算法。

第四章提出了移动机器人调度算法并且将该算法与一种传统的物流仓库订单分拣算法进行了比较，利用仿真结果进行数据对比，得出结论。

第五章对全文进行了总结，总结了所做工作的不足之处，并且对未来进行了一定的展望。

1.5 本章小结

本章首先介绍了课题的研究背景及其意义，之后介绍了国内外在仓库订单分拣方面的研究现状，最后对全文的主要研究内容进行了一个概括。

第 2 章　问题描述及定义

2.1　问题描述

在物流中心的拣货操作的目标是满足包括要求物品，数量，及期限在内的用户需求。每天，仓库的决策者都会收到很多用户订单，并且必须制定拣货和运输计划以满足客户需求。每份用户订单由很多物品组成，而这些物品分散在仓库中的特定位置中。

一般物流仓库中订单分拣存在两种方式：

整体分拣，仓库中货架较小，可以由底部的移动机器人进行驱动，根据接收到订单的货物要求，自动运送货架到达仓库工作人员所在的打包区，工作人员每次处理一个订单，照单从货架上取货，然后打包送出。

非整体分拣，在这种方式下，订单中的多种货物完全分开独立处理。使用流水线或轨道式移动机器人可将货物快速移动出来并自动化的打包、贴标签、装车。

但是一般以上的分拣方式均需要一定程度的人工参与，难以实现完全自动化，为此，提出了可以完全依靠货架，移动机器人进行订单分拣的物流仓库模型。物流仓库在接受到外部订单后，需要根据货物所在位置，将其分配给特定的机器人去完成订单，同时也需要考虑到机器人在移动过程中在路径上的可能冲突。

在传统结构的物流仓库中，由于货架摆放位置，不管是是人工还是机器人分拣，均需要根据订单中货物的位置，制定进入仓库的路径，使得分拣效率最高，这就是传统的路径规划问题。但是，在本文的模型中，无需考虑路径规划，只需要考虑货物分配给哪个机器人使得行驶路径最短。

2.2　基本定义

现将物流仓库模型定义入下：

（1）所有货架均为智能货架，具有如下能力：

货架一侧陈列货物，可接收需要货物的信息，并将该货物取下放置到其下方的无人车上（实现方式可为推取式或机械手等）。

货架另一侧可接收订单信息，并在本订单的所有货物到齐后打包送出。

每一货架只陈列同一种类的物品，分为上下多层，机械手可根据接受的订单

货物信息，将所需物品所在层的货物准确推下。

（2）若干移动机器人（无人车），具有如下能力：

可自动运送并装卸货物。

（3）中央处理机构，具有如下能力：

获知所有无人车的运行状态，包括当前的运动学信息，包括位置、速度、加速度等。如果车辆处于运输货物状态中，则可获知所运输货物的订单信息。

规划每辆车运行方式的能力。

对每个订单发放到一个货架上打包的分配能力。

对订单中货物进行分配的能力。

除此之外，该特定结构的物流仓库的货架摆放方式亦不同于传统的物流仓库，货架连续摆放成封闭的正方形，移动机器人（无人车）在货架围成的正方形内运输货物，而货物到达并打包后输出到货架外侧由机械手运送到传送带上分目的地发往各个地方。

2.3 本章小结

本章主要对本文所研究的问题进行了描述，阐述了研究机器人调度算法与传统物流仓库的路径规划算法的不同之处，之后对本文提到的仓库的各个硬件组成部分进行了定义，描述了它们各自需要具备的功能。

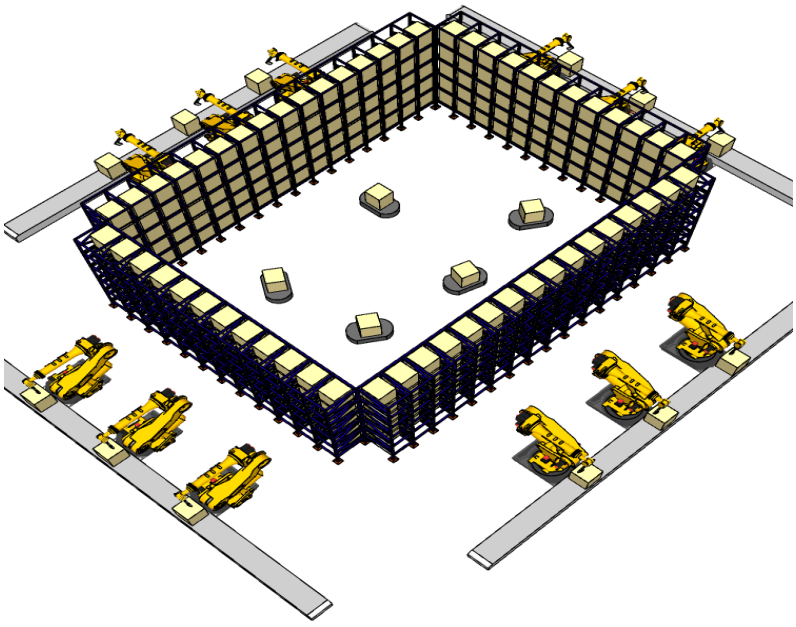

图 2-1 物流仓库立体图

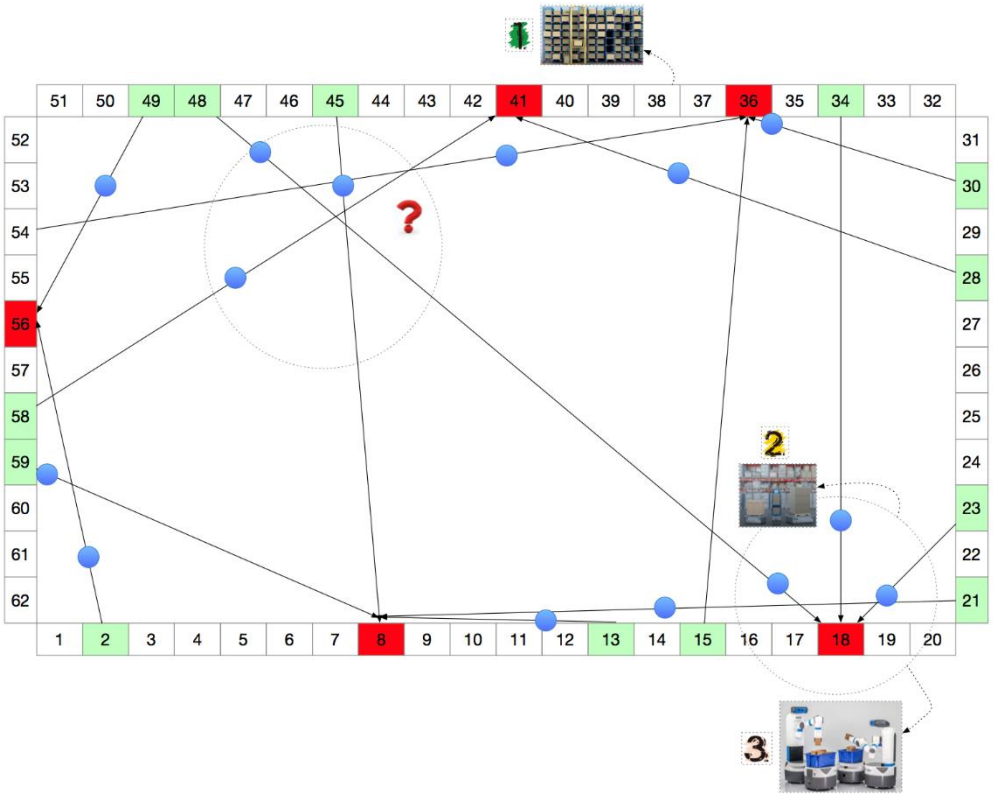

图 2-2　物流仓库平面示意图

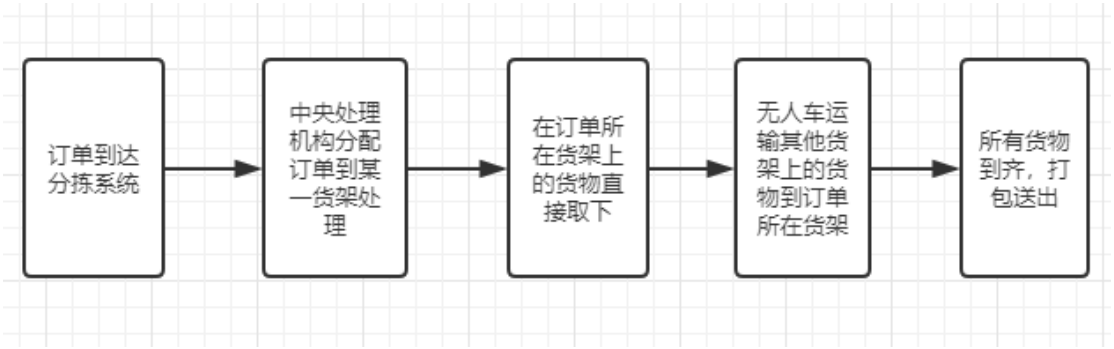

图 2-3　物流仓库订单分拣流程

第 3 章　软件平台及算法

3.1　MATLAB 上的软件仿真平台

为了实现物流仓库订单分拣全部流程的仿真，我们在 MATLAB 上搭建了软件仿真平台。

该仿真平台拥有简单的图形界面，可以直观的显示物流仓库中由移动机器人进行订单分拣的全部过程，界面如图所示。图中边缘方框代表货架，一共有 88 个，其中每个方块的边长是两个单位长度，所有货架围成了 48x48 的正方形区域；红色的圆点代表移动机器人（无人车），直径大约为一个单位长度，初始随机停放在不同货架下；当系统接收到订单后，会用深蓝色五角星在货架上标注该订单中需求的货物，并且将最终打包点货架用淡蓝色的五角星标记。移动机器人会分别将订单中的需求货物运送到最终打包货架，并且货架在货物全部到齐后将其打包送出。移动机器人承担运输任务时，会在起点与终点之间标记一条线表示行进线路，在到达目标地点后该行进路线会消去。关于该仿真平台的具体算法实现，将在下面介绍。

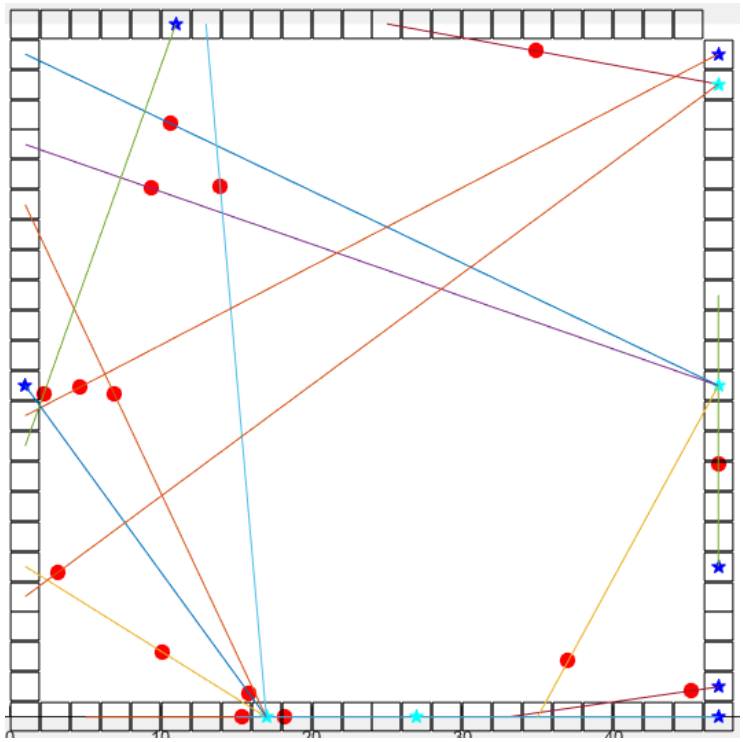

图 3-1　物流仓库仿真模拟图

3.2　各个模块功能

3.2.1 接受订单

物流仓库每一定时间内收到的客户订单数量一般满足服从指数分布的函数。指数分布的概率密度函数一般表述如下：

$$f(x)=\begin{cases}\lambda e^{-\lambda x} & x>0\\ 0 & x\le 0\end{cases} \tag{3-1}$$

其中 λ > 0 是分布的一个参数，常被称为率参数。即每单位时间内发生某事件的次数。

指数分布的分布函数为：

$$F(x;\lambda)=\begin{cases}1-e^{-\lambda x} & x>0\\ 0 & x\le 0\end{cases} \tag{3-2}$$

那么订单到来假如符合 λ=mu 的指数分布，则每一单位时间内有订单来到的概率为：

$$\mathrm{P(X\le 1)=F(X)}=1-e^{-mu} \tag{3-3}$$

除此之外，每份订单中的需求货物数量均有一个上限，其数量在该上限内随机均匀分布；同时，每份订单中需求货物所位于的货架亦是随机均匀分布的。这就保证了每一份订单的完全随机性。

3.2.2 移动机器人

移动机器人在货架围成的正方形内部运行，所有机器人均受到中央处理机构的统一调度。为了达到最高效率，它们从任何起点到目标点均只沿直线行走。

移动机器人拥有 3 个状态，分别是空闲，拿取货物和运送货物。

空闲状态是机器人没有被分配订单，处于静止状态；拿取货物状态是机器人接受订单后，前往货架拿取货物的过程中的状态；运送货物状态是机器人拿到指定货物之后前往打包货架过程中的状态。

3.2.3 预测控制

通常情况下，为了提高订单分拣的效率，物流仓库中的移动机器人都多于一个。在此模型中也不例外，但是复数移动机器人在封闭的有限面积区域内在不同货架之间沿直线移动，必然会出现路线互有交叉，以及多个移动机器人相互碰撞的现象。为了解决这一问题，该实验平台采用了预测控制算法，由中央处理机构获知每个移动机器人的位置，速度，加速度等信息，通过预测控制计算，统一控

制所有的移动机器人，使得它们不至于相互碰撞。以提高订单分拣的效率。

3.2.4 订单发放及分配

物流仓库在接收到用户的订单后，一般会按照订单的紧急程度进行优先度排序，将优先度比较高的，或是截止时间相同的订单一起处理，以提高订单分拣效率。但是，为了简化，我们默认订单的处理方式为先到先处理，并且只有在一个订单中的货物全部分配给移动机器人之后才进行下一个订单的分配。

在本模型中，仓库接收到的订单需要先发放到某一个货架上进行处理，再根据订单中货物所在的位置将它们分别分配给周围的临近处于空闲状态的机器人，由机器人运送货物到处理货架上，再一起打包送出。

仓库接收到的订单有三种状态：未分配状态，正在处理状态和已完成状态。

未分配状态是接收到订单，但还未将其发放到某一货架；正在处理分配状态是订单已被发放到货架上，其中的每个货物正在被分配给移动机器人去处理；已完成状态是指订单中的每个货物均已经由移动机器人运送到打包货架，可以打包送出。

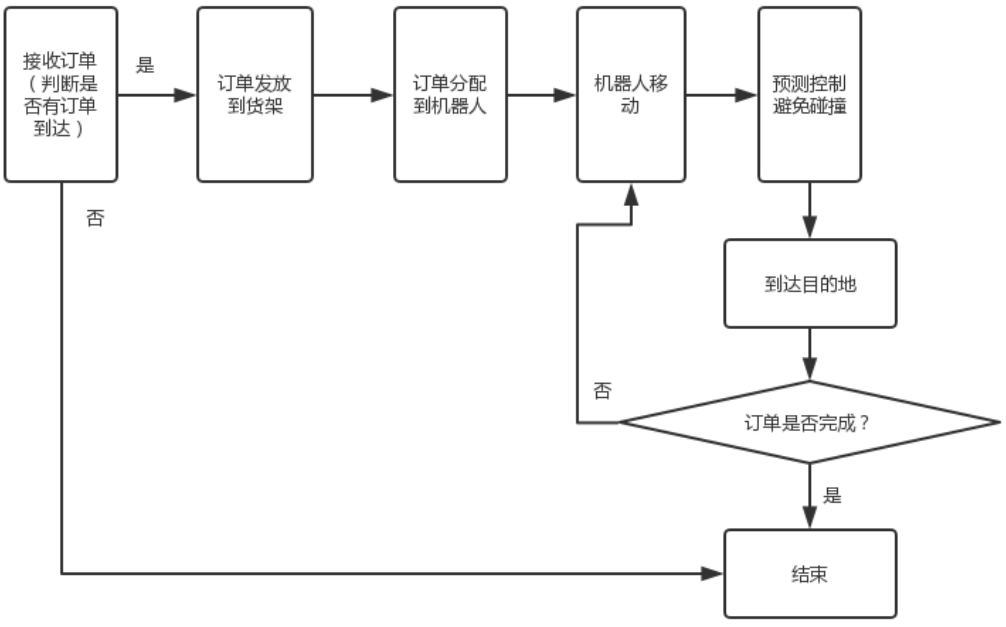

图 3-2　各个模块之间的关系流程图

各个模块之间的关系如图 3-2 所示，系统在接收到订单之后，将订单发放到货架，中央控制系统根据订单中货物的位置将货物分配给临近的机器人运输，机

器人移动到目的地，在途中应用预测控制算法避免碰撞，完成订单后再执行下一个订单。

3.3 本章小结

本章主要对物流仓库的 MATLAB 仿真平台进行了详细的介绍，对软件平台中的各个模块功能进行了一一介绍，并且详细阐述了物流仓库接受订单，分拣订单，到打包送出的工作流程以及各个模块之间的相互关系。

第 4 章　多移动机器人调度算法及其与传统算法的比较

4.1　算法介绍

本算法是针对前文介绍的特殊结构的物流仓库的订单分拣而提出的，主要包括订单发放，订单分配和预测控制算法，下面将对该算法进行详细介绍。

订单发放是指当物流仓库接收到一份订单，需要根据订单中所包含的货物及其所在货架位置，来决定选取哪个货架作为最终打包货架，即这份订单中的所有货物均由移动机器人从其他货架上运输到这一货架，等到齐后一起打包送出，从而使得这份订单完成的时间最短。

订单分配是指对于一份订单中的每一个货物，如何将它分配给有限的移动机器人，使得订单中货物的总运输时间最短。

为了是每一份订单均具有最小的完成时间，即尽可能快的完成订单，我们需要找出使得订单中货物所在货架到达具有最短路径和的打包货架和距离每个货架最近的机器人。

在机器人在仓库内移动的过程中，由于行驶路线均是一条直线，因此前往两个不同目的地的机器人间难免发生路线交叉，甚至碰撞的现象，并且随着移动机器人的增加出现的概率大大增加。这时就需要根据每个机器人的当前位置和运动状态来预测下一步它所处的位置，以提前做出避让。

4.2　算法表达式

4.2.1　订单发放及分配

物流仓库每接收到一份订单，就必须尽快将它发放到仓库中的一个货架上，将其作为最终打包货架，位于其他货架上的货物均由移动机器人运输到该货架，再一起打包送出。

由于移动机器人只在起点和终点之间沿直线移动，且其运动时的最大速度相同。因此为了使得每份订单能在最短的时间内完成，可以只需要考虑订单中货物所在货架与移动机器人之间的最短距离，再加上它们与某一货架之间的距离最短就可以了。

集合，参数,变量说明。

集合

L ： 存储位置集合 L={1,…,l}

L_n:第 n 个订单包含货物所在的货架位置集合

C ： 移动机器人集合 C={1,…,c}

N ： 订单集合 N={1,…,n}

M ： 订单中的货物集合 M={1,…,m}

参数

C_l:距离货架 l 最近的空闲移动机器人位置 $C_l \in C$

L_{mn}:第 n 个订单中第 m 个货物所在的货架位置

d(i, j)：在位置 i 和 j 之间的行驶时间

F_n ： 第 n 个订单的最终打包货架 $F_n \in L$

β^p ： 每件物品的拣取和搜索时间

B_n ： 订单 n ∈ N 需求的物品数量

变量

$\mathbf{T_n}$ ： 订单 n 的完成时间

X_{cmn} ：表示订单 n 中的货物 m 被分配到移动机器人 c，当且仅当订单 n 中的货物 m 被分配到移动机器人 c 时等于 1

Y_c ： 当且仅当移动机器人 c 中只有一个货架上的货物时等于 1

R_{in} ： 表示第 n 份订单中的第 i 个货物被分配，当且仅当第 n 份订单中的第 i 个货物被分配时等于 1

目标函数：$\text{Min} \sum_{i \in L_n, i \neq F_n} d(L_{in}, C_j) + \sum_{j \in L_n} d(L_{jn}, F_n)$ (4-1)

约束：

$$\sum_{m \in M} \sum_{c \in C} X_{cmn} = 1 \quad (4\text{-}2)$$

$$Y_c = 1 \qquad \forall c \in C \quad (4\text{-}3)$$

$$\sum_{i \in M} R_{in} = m \quad (4\text{-}4)$$

$$\mathbf{T_n} \geq B_n \cdot \beta^p + \sum_{i \in L_n, i \neq F_n} d(L_{in}, C_j) + \sum_{j \in L_n} d(L_{jn}, F_n) \quad (4\text{-}5)$$

目标函数(4-1)最小化了订单完成时间。约束条件(4-2)确保一个货架上的货物只能分配给一个移动机器人，约束条件(4-3)确保一个移动机器人只能一次只能运输一个货架上的货物。条件(4-4)确保了每一份订单中的货物都必须被分配。条件

(4-5)中，确定了一份订单总的完成时间。

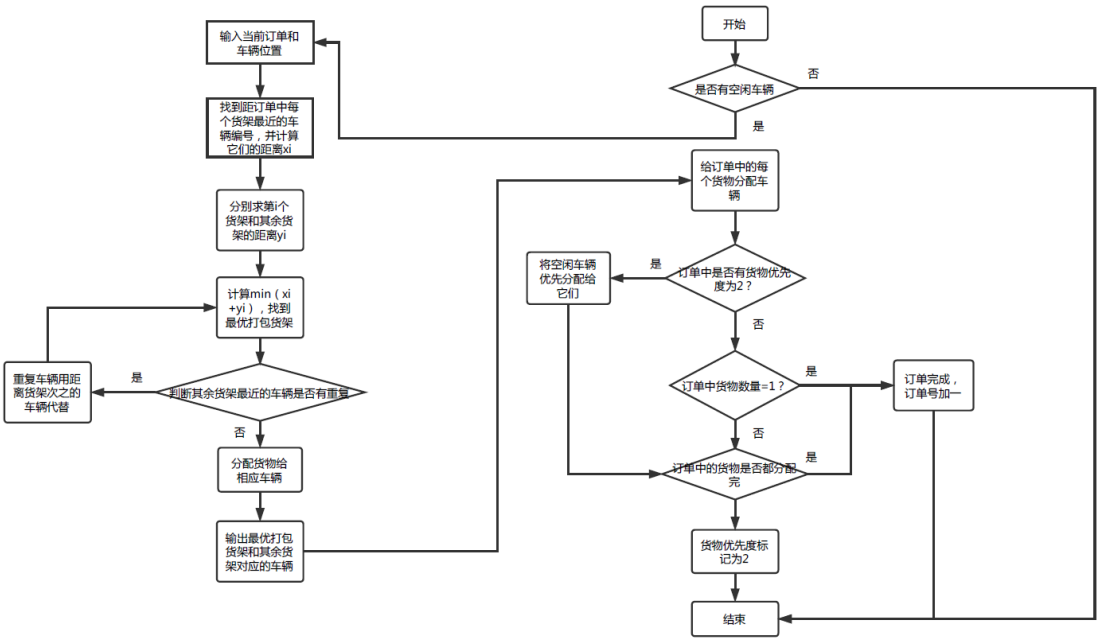

图 4-1　订单发放和分配流程图

4.2.2　预测控制

移动机器人在一定面积的仓库内运动过程中，难免会出现行进路线的交叉以及碰撞的现象，为了避免这种现象，使得机器人能够无碰撞的完成货物运输任务，需要中央控制系统根据当前所有机器人的位置以及运动信息，及时作出判断，得到下一步甚至之后几步的机器人位置信息，计算出加速度，以提前做出避让。预测控制的状态空间模型如式(4-6)，(4-7)所示：

$$\begin{bmatrix} S_x(k+1) \\ S_y(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \triangle t\cos\theta \\ 0 & 1 & \triangle t\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x(k) \\ S_y(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(\triangle t)^2\cos\theta \\ \frac{1}{2}(\triangle t)^2\sin\theta \\ \triangle t \end{bmatrix} a \quad (4\text{-}6)$$

$$\begin{bmatrix} S_x(k) \\ S_y(k) \\ v(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x(k) \\ S_y(k) \\ v(k) \end{bmatrix} \tag{4-7}$$

预测控制的计算表达式如下

参数，变量定义如下：

参数：

L:表示机器人的直径

V:机器人速度

v_0：机器人初速度

a:机器人加速度

S_y:机器人沿 y 轴方向的位移大小

S_x：机器人沿 x 轴方向的位移大小

n：机器人数量

θ：机器人运动方向和 x 轴的正向夹角

$S_y^i(k+m)$:表示第 k+m 时刻第 i 个机器人沿 y 轴方向的位移分量

$S_x^i(k+m)$:表示第 k+m 时刻第 i 个机器人沿 x 轴方向的位移分量

变量：

ε_i：二值变量，当且仅当第 i 个机器人空闲时为 1，不空闲时为 0

λ_i：第 i 个机器人的优先度

Δt:中央处理器的采样时间

目标函数：$\min \sum_{i=1}^{n} \varepsilon_i \frac{v_{i,0}+a_i\Delta t}{\lambda_i}$ (4-8)

约束条件：$S_{y,min} \le S_y + v_0\Delta t + \frac{1}{2}a\Delta t^2 \le S_{y,max}$ (4-9)

$$S_{x,min} \le S_y + v_0\Delta t + \frac{1}{2}a\Delta t^2 \le S_{x,max} \quad (4\text{-}10)$$

$$v_{min} \le v_{i,0} + a_i\Delta t \le v_{max} \quad (4\text{-}11)$$

$$a_{min} \le a \le a_{max} \quad (4\text{-}12)$$

$$\sqrt{(S_y^i(k+m)+\sin q^i \times s_m - S_y^j(k+m)+\sin q^j \times s_m)^2 + (S_x^i(k+m)+\cos q^i \times s_m - S_x^j(k+m)+\cos q^j \times s_m)^2} \ge R \quad (4\text{-}13)$$

其中：

$$\sigma_m = vm\Delta t + \frac{1}{2}a(m\Delta t)^2 \quad (4\text{-}14)$$

$$R = \sqrt{L^2 + (v_{\max}\Delta t)^2} \quad (4\text{-}15)$$

目标函数(4-8)规定了单步性能指标，约束条件(4-9)表示机器人沿 y 方向的位移范围，约束条件(4-10)表示机器人沿 x 轴方向的位移范围，约束条件(4-11)表示机器人末速度的范围，约束条件(4-12)表示机器人的加速度范围，约束条件(4-13)表示机器人避免碰撞应保持的最小距离。

4.3 传统的物流仓库模型

传统物流仓库一般包含一个手动的，低级别的“货到人”订单拣配系统，从这个系统中必须取得用户订单中指定的货物。这些货物一般被存放在直接能被拣货员接触到的仓库中的货架上。货物的存储位置构成了一种典型的由所谓的拣选过道和交叉过道组成的区块布局。 拣货通道相互平行，包含了布置在每个拣货通道两侧的存储位置。交叉通道不包含任何存储位置，但能使拣货员进入或离开拣货通道。此外，交叉通道将拣货区域分成几个区域，并将拣货通道分成一些子通道。一个区块由位于两个相邻的交叉通道之间的拣货区形成。拣货通道的相应部分被表示为子通道。因此，包含 m 个拣货通道和 q + 1 个交叉通道的仓库包括 q 个区块和 q•m 个子通道。此外，仓库还包含一个入口，拣货员可以由此进入拣货区域，以便拣配物品，并且在拿齐了要求的货物之后返回，将货物按订单二次

分拣进行打包，之后由货车运输到用户处。

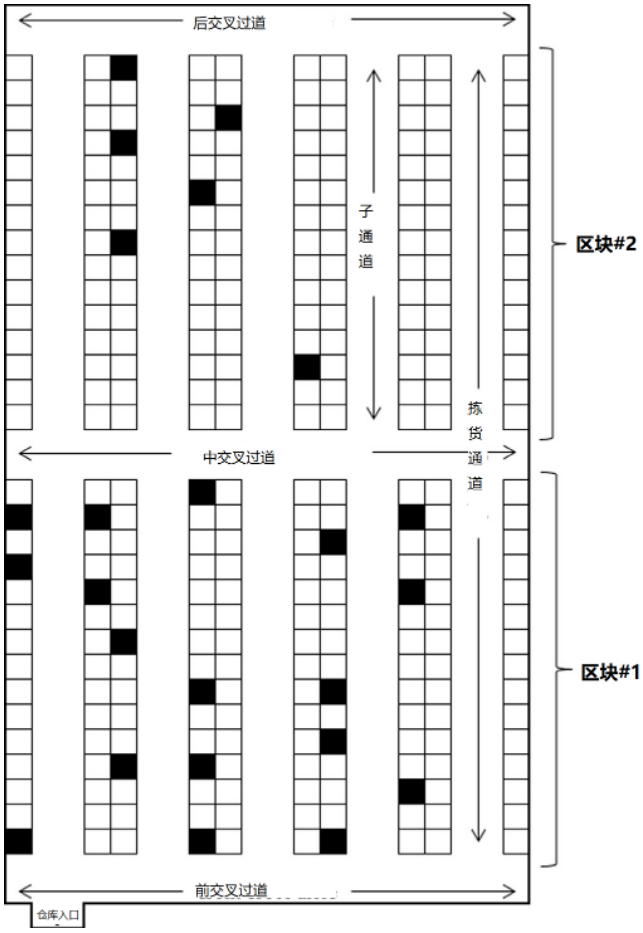

图 4-2　传统物流仓库平面图

在对比实验中，考虑到仓库中货架数保持不变，及仓库中所能存放的货物数量，种类不变，以保证两个仓库模型都只涵盖相同数量和种类的货物。我选取了具有两个块和六个拣选通道的拣货区域。两个区块布局的特点是有三个交叉通道，即前，中，后交叉通道，前交叉通道和后交叉通道分别代表最靠近和最远离仓库入口的交叉通道。中间的过道将两个区块彼此隔开。存储位置由矩形表示，而黑色矩形表示需求物品的位置（拣货位置）。

4.4　传统路径规划算法

S 形策略是实践中最常用的路径规划算法。当应用这个算法时，订单拣选员遍历每个包含至少一个要求物品的子通道。订单拣选员将从包含需求货物的最左

边的拣选过道穿过仓库后部，然后返回到前面按区块拣货。不含有货物的区块将被跳过，在选取某一区块中的最后一个货物后，订单拣货员将返回到该区块的前面并继续下一个区块。直到拣选完一个批次订单中的货物后返回仓库入口。

蚁群算法也是一种常用的路径规划算法，该算法中将 S 只蚂蚁放在随机节点上，蚂蚁在初始状态期间随机选择要访问的位置，因为每个存储位置之间的路径尚未累积信息素。每只蚂蚁不会选择出现概率最高的地点。而是生成一个随机数，并使用概率方法来选择下一个存储位置。当所有蚂蚁走完所有节点，就会根据所走的路径产生信息素，根据信息素的浓度决定下一代的蚂蚁的行走路径的概率。

组合策略算法是一种更复杂的算法。它结合了 S 形策略和回归策略的元素，它与 S 形策略有相同的访问顺序。但是，该策略能穿越子通道并在该子通道中转向。这些选择是通过动态编程来完成的，从某种意义上说，它可以提前预示一个子通道。例如，假设对于一个子通道来说，返回是最短的路径，但是如果它能为下一个子通道提供更好的起点，那么组合策略可以选择让拣货员直接穿过该子通道。

4.5 仿真及对比结果

本仿真实验即对比实验在使用 Windows10 专业版系统，配备了 Intel Core 处理器 i3-6100 3.70GHz 和 4.00GB RAM 内存的个人电脑上进行。MATLAB 的版本为 2016B。对比实验采用了最常用的 S 形策略算法，同样在 MATLAB 上搭建了软件平台，采用与本文的软件平台相同数量的货架，货架排放采用两个块和六个拣选通道的拣货区域。仿真结果和数据如表所示：

mu:表示 λ=mu 的指数分布

N:表示预测控制的预测步数

C:表示仓库中的车辆数

L:表示主程序循环次数

表格 4-1：

表格 4-1　L=200　mu=0.3　每个订单的货物数小于等于 10

	收到订单数（个）	完成订单数（个）	运送货物数（个）	运行总时间（秒）	堵塞总次数（次）

		平均		平均		平均		平均		平均	
N=1	C=10	47	45.67	2	3.67	23	26	148.89	126.61	14	8.33
		49		4		27		108.52		3	
		41		5		28		122.43		8	
	C=15	46	48.67	5	6	49	40.33	191.80	179.87	7	6.33
		49		5		35		170.86		7	
		51		8		43		176.96		5	
	C=20	49	55	7	8	49	51.33	228.91	224.92	14	11.67
		56		8		56		235.61		11	
		60		9		49		210.24		10	
	C=30	47	50	7	10.67	74	70.67	365.48	356.37	30	27.33
		49		13		63		329.26		26	
		54		12		75		374.36		26	
N=2	C=10	46	48.67	4	3.33	30	28	295.64	288.72	3	4
		49		3		26		285.17		5	
		53		3		28		285.36		4	
	C=15	50	51	4	5.33	37	38	374.07	374.15	5	5
		55		6		36		348.94		2	
		48		6		41		399.45		8	
	C=20	48	49.33	4	2.6	38	40	556.21	572.51	13	8.33
		54		4		39		587.10		9	
		46		5		43		574.21		3	
	C=30	41	49.67	9	4.33	66	54.33	851.59	923.89	13	7.67
		48		3		49		933.89		9	
		60		1		48		986.19		11	

N=3	C=10	57	54.33	5	5.33	32	33.67	388.82	383.82	5	2.67
		59		5		29		356.49		3	
		47		6		40		406.15		0	
	C=15	52	48	7	6.33	42	40.67	598.04	638	4	6
		45		2		31		655.34		8	
		47		10		49		660.62		6	
	C=20	59	51	3	3.33	47	44	997.81	993.93	20	12.33
		50		4		43		923.09		9	
		44		3		42		1060.89		8	
	C=30	49	49.67	9	6.33	66	68	1775.14	1845.75	9	
		46		5		64		1772.75			
		54		5		74		1989.37		22	
N=4	C=10	49	48.33	4	3	30	29	648.89	624.04	6	5
		48		2		26		646.80		7	
		48		3		31		576.42		2	
	C=15	50	45.33	6	5.33	39	44.67	939.12	968.82	5	4.67
		47		5		47		995.87		2	
		39		5		48		971.48		7	
	C=20	49	52	9	8	53	52.33	1465.78	1505.51	9	8.33
		52		6		54		1510.92		8	
		55		9		50		1539.84		9	
	C=30	58	51	13	11.33	90	92.33	2446.65	2706.32	8	8.33
		48		8		93		2851.93		9	
		47		8		94		2820.39		9	
N=5	C=10	50	48.67	4	4	25	25.33	564.64	574.36	3	2.33
		47		1		17		590.46		2	

		49		7		34		567.98		3	
	C=15	50	48.67	4	5.33	28	40.33	1369.14	1314.01	6	7.67
		50		4		38		1242.86		5	
		46		8		55		1330.03		12	
	C=20	48	50.33	8	7.33	68	60.67	2202.08	2158.87	5	5.67
		57		8		64		2182.34		4	
		49		7		50		2092.18		8	
	C=30	44	49.33	19	12.67	124	105	4242.34	4207.80	10	6.33
		55		8		97		4390.30		5	
		49		11		94		3990.75		4	
N=6	C=10	46	50	1	3	23	30	1054.10	1014.04	5	3
		48		2		30		1118.50		3	
		56		6		37		869.52		1	
	C=15	50	47.67	9	7.33	46	48	1560.77	1576.46	3	4.33
		50		8		49		1616.71		2	
		43		5		49		1551.90		8	
	C=20	50	53.67	8	8.67	65	69.67	2865.97	2951.49	5	3.67
		57		9		68		3125.70		4	
		54		9		76		2862.81		2	
	C=30	47	49.67	10	12.33	102	109.33	5709.40	5922.29	7	7.33
		54		17		125		6352.26		8	
		48		10		101		5705.20		7	

在主程序循环一定的次数的情况下，仓库中订单的完成数量一般取决于：移动机器人的数量，每份订单中包含的货物数，仓库大小等因素。由于每份订单中的货物数量是随机的，只存在上下界，因此比较机器人运送货物的数量来判断运输效率。从表格 4-1 中可以看出，

⑴当一定面积的仓库内移动机器人数量越多，其运送的货物也就相应增多，而且一般来说完成订单个数也相应增加。

⑵随着移动机器人数量的增加，其发生堵塞的概率也会加大，导致堵塞次数呈增长趋势。

⑶应用预测控制算法时，随着预测步数的增加，在移动机器人数量较少时，运送货物数量与堵塞次数均无明显变化。但是在移动机器人数量较多时，可以明显看到运送货物数量有增加趋势，堵塞次数有减少趋势。

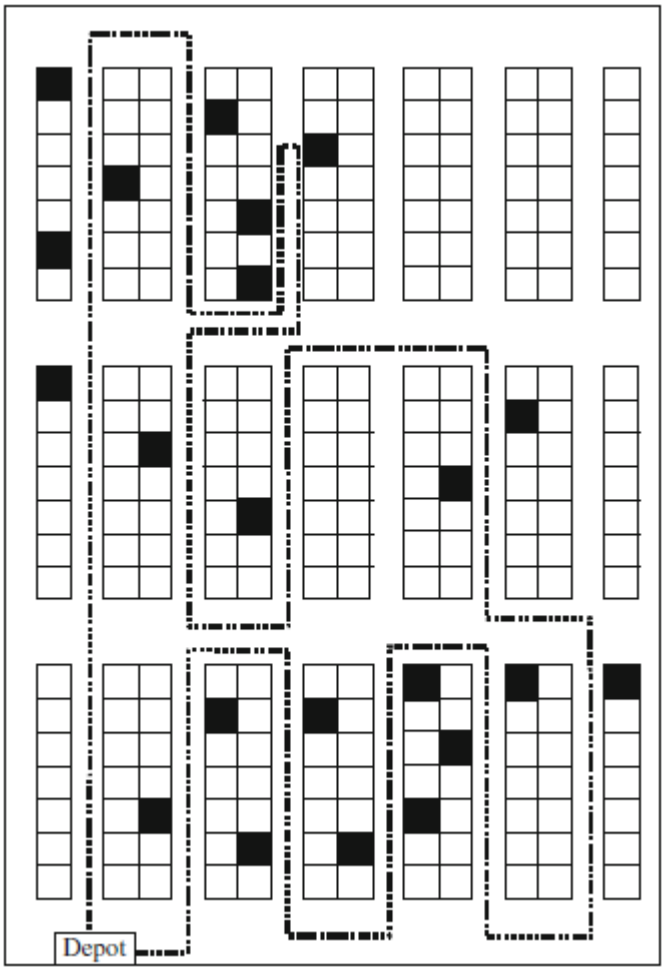

图 4-3　S 形路径规划算法

表格 4-2：

表格 4-2　　L=200　　mu=0.3　　每个订单的货物数小于等于 10

	收到订单数（个）	完成订单数（个）	拿取货物数（个）	运行总时间（秒）	堵塞车辆数（个）

	平均		平均		平均		平均		平均	
C=10	47	48.33	2	2.33	31	39.67	320.75	330.26	6	6.33
	51		1		42		357.32		7	
	47		4		46		312.71		6	
C=15	46	44.33	3	2	30	30	471.22	407.03	10	9.33
	41		2		34		344.46		5	
	46		1		26		405.40		13	
C=20	62	56	6	2	43	48.33	547.89	557.48	8	10.33
	58		0		55		562.98		13	
	48		0		47		561.56		15	
C=30	51	55.33	0	1	66	64.67	783.95	788.01	18	19
	53		3		66		728.85		22	
	62		0		62		851.23		17	

由表格 4-2 和表格 4-1 对比可以看出，在预测一步时，当仓库中有 10 个移动机器人，传统模型中机器人拿取的货物数要多于本文的模型，但是当机器人数量增加到 15 个及以上时，传统模型明显不及本文中的模型，无论是完成订单个数还是运送（拿取）货物的数量。

由此可以得出本文提出的模型要优于传统物流仓库模型的 S 形路径规划算法的结论。

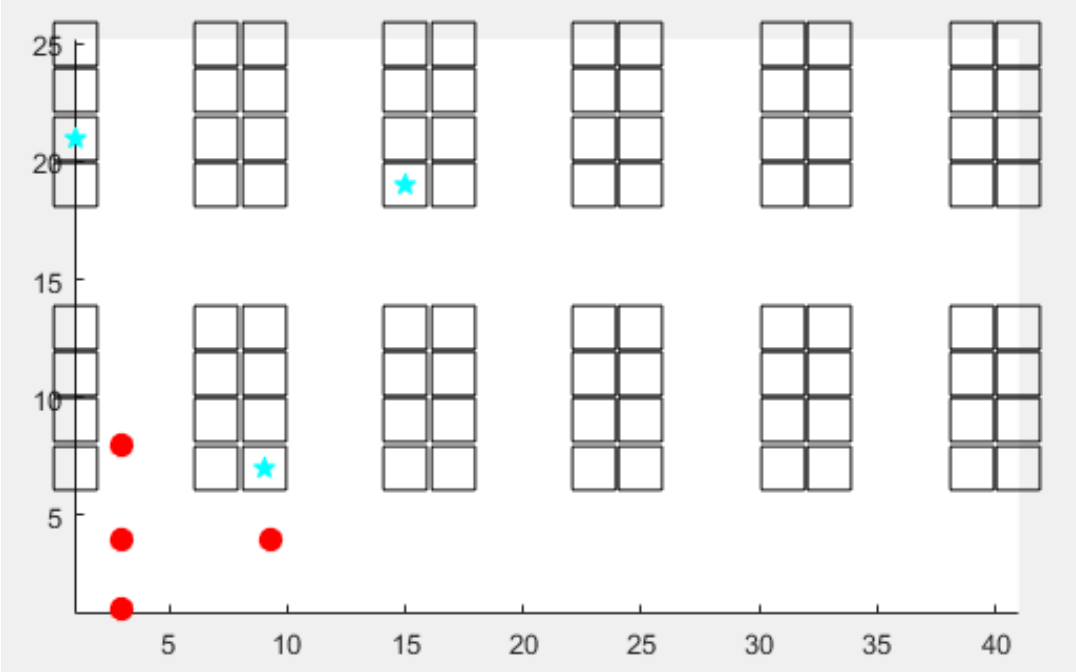

图 4-4　传统物流仓库仿真运行图

4.6　本章小结

本章着重介绍了订单发放和订单分配的算法及思想，并且将其应用在所搭建的 MATLAB 仿真平台上，进行了性能测试并且得出了结论。之后介绍了传统的物流仓库订单分拣方法以及相应的路径规划算法，选取了应用 S 形策略的路径规划算法的传统物流仓库模型，在 MATLAB 上搭建仿真平台，对这两种模型在一定订单到来的概率下进行了对比测试，得出结论。

第 5 章　总结和展望

5.1　毕设工作总结

电子商务是一种新兴的基于网络的商业运营模式,买卖双方在网络上确定交易活动,然后通过某种运输方式将所交易的货物运送到买家手中。随着互联网以及智能终端的普及，人们越来越青睐于通过网络平台订购商品。随之而来的是大量的订单，以及对物流仓库订单分拣和物流运输能力的考验。为了提升对大批量订单的处理能力，提高订单分拣的效率，同时降低成本，拥有智能仓储物流机器人的智能自动化仓库越来越普遍。它能有效降低人力成本，减少出错，提升订单处理效率。

本文的研究内容是基于特定物流仓库模型的面向订单分拣的多移动机器人调度算法。现将本文的工作总结如下：

⑴在阅读许多有关订单分拣的文献的基础上，介绍了课题的背景以及研究意义，简单总结了目前国内外对于订单分拣问题的研究现状。

⑵介绍了本文所基于的特定结构的物流仓库模型以及以此为模型在MATLAB 上搭建的软件仿真平台和它的主要组成部分。

⑶详细介绍了本文的主要研究内容，基于特殊物流仓库结构的面向订单分拣的多移动机器人调度算法。

⑷简单总结了传统的物流仓库模型以及常用的订单分拣的路径规划算法，并选取了其中一种在 MATLAB 上进行仿真，并与本文所列算法进行对比，比较它们最终的订单处理效率，本文的算法在综合意义上优于该传统算法。

5.2　未来展望

随着毕业设计课题的深入研究，越来越深刻地意识到本文研究内容的不足。由于本人的时间及能力有限，目前的课题内容研究还处于初级阶段，还有许多需要改进的地方：

⑴本文对于物流仓库接收到的订单，只是简单地按照先到先分配的原则进行处理，而未考虑可能存在某些订单用户对于处理时间有特殊的要求，比如十

分紧急的订单。这就需要对订单依据截止时间进行优先度分级，对于高优先度的订单先进行分配，而将低优先度的暂缓处理。

⑵由于物流仓库中的移动机器人统一归中央处理机构管理调度，中央处理机构为时时获取每个机器人的信息，需要以一定的频率进行数据传输。为了简化模型，本文没有考虑机器人与中央处理机构的信息传输问题，比如信息传输的频率，如果信息丢失怎么办，信息传输延迟对系统的影响等等。

⑶由于采用了多步预测控制的方法来防止移动机器人之间产生碰撞，大大增加了整个仿真程序的计算量，使得计算时间随着预测步数增加成倍增长，计算开销比起传统的算法大了不少。

参 考 文 献

[1] Chen, T.-L. , Cheng, C.-Y. , Chen, Y.-Y. , & Chan, L.-K. An efficient hybrid algorithm for integrated order batching, sequencing and routing problem[J]. International Journal of Production Economics, 2015 ,159:158–167 .

[2] AndréScholz ,Daniel Schubert,Gerhard Wäscher, Order picking with multiple pickers and due dates – Simultaneous solution of Order Batching, Batch Assignment and Sequencing, and Picker Routing Problems[J]. European Journal of Operational Research, 2017,263:461–478

[3] Henn, S. Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses[J]. Flexible Services and Manufacturing, 2015,27 : 86–114 .

[4] Henn, S. , Koch, S. , & Wäscher, G. Order batching in order picking ware- houses: A survey of solution approaches. In R. Manzini (Ed.), Warehousing in the global supply chain: Advanced models, tools and applications for storage systems [M] (pp. 105–137). London: Springer ,2012.

[5] Tsai, C.-Y. , Liou, J. J. H. , & Huang, T.-M. Using a multiple-GA method to solve the batch picking problem: Considering travel distance and order due time[J]. International Journal of Production Research, 2008 ,46 :6533–6555 .

[6] 刘进平.配送中心订单分拣系统中的拣货路径选择研究[J]. 运筹与管理. 2010(05)

[7] 韦超豪. 面向 B2C 电商平台的订单分拣优化研究[D].合肥工业大学 2016

[8] 潘成浩. 仓储物流机器人拣选路径规划仿真研究[D].中北大学 2017

[9] Wäscher, G. Order picking: A survey of planning problems and methods. In H. Dyckhoff, R. Lackes, & J. Reese (Eds.), Supply chain management and reverse logistics [M] (pp. 323–347). Berlin: Springer, 2004.

[10] Scholz, A. , Henn, S. , Stuhlmann, M. , & Wäscher, G. A new mathematical programming formulation for the single-picker routing problem[J]. European Journal of Operational Research, 2016,253 : 68–84 .

[11] 马廷伟. 物流中心订单分拣策略的研究[D].北京邮电大学 2015

[12]王宏琴,贾晓庆. B2C 企业自建物流中心分拣系统研究[J]. 中国市场. 2012(28)

[13]刘田. 仓储系统中分拣、存取效率优化模型与策略研究[D]. 华中科技大学 2016

[14]王文蕊. 电子商务配送中心的设计与优化策略研究[D]. 山东大学 2014

[15]肖际伟. 配送中心拣货系统优化[D]. 山东大学 2010

[16] 顾彦. 科技正在改变物流行业[J]. 中国战略新兴产业. 2016(25):45-47.

[17]马超, 雷斌, 陈洪满. 一种配送中心订单分拣优化问题的研究[J]. 吉林师范大学学报(自然科学版), 2014(3):118-121.

[18]郭进. 多订单并行分拣问题的优化研究[D]. 上海交通大学, 2012.

[19]Roodbergen K J. Storage Assignment for Order Picking in Multiple-Block Warehouses[J]. Warehousing in the Global Supply Chain, 2012:139-155.

[20]刘同娟, 马向国. 配送中心不同分拣策略的仿真优化[J]. 物流技术, 2013, 32(5):439-444.

[21]Koster R D, Le-Duc T, Roodbergen K J. Design and control of warehouse order picking: A literature review[J]. European Journal of Operational Research, 2007, 182(2):481-501.

[22]Hong S, Kim Y. A route-selecting order batching model with the S-shape routes in a parallel-aisle order picking system[J]. European Journal of Operational Research, 2017, 257(1):185-196.

[23]Duc T L. Design and control of efficient orderpicking process[J]. 2017.

[24]宋宝瑞. 配送中心分拣系统运作分析[J]. 中国证券期货, 2012(4):253-253.

[25]刘进平. 配送中心分拣系统设计方法研究[D]. 大连海事大学, 2011.

附录

#主程序

```
load car;
load huojia;
%load order;
set(gcf,'unit','centimeters','position',[10  5 20 13]) % 框图大小
set(gca,'Position',[.15  .15 .8 .75]);
set(get(gca,'XLabel'),'FontSize',8);
global CARNUM t; % 初始化数据
global vnum xnum ynum sinnum cosnum movenum max;
CARNUM=15;%移动机器人数量
HUOJIANUM=92;%货架数量
m=1;%表示第 m 个订单
n=1;%表示第 m 个订单中的第 n 个货物
t=1;
axis equal
hold on
%移动机器人结构体，包含状态，当前坐标，目的地坐标，速度，加速度，处理订单编号等信息
for i=1:CARNUM
ncar1(i)=struct('state',0,'move',0,'num',car(i,1),'x',car(i,3),'y',car(i,4),'begiX',0,'begiY',0
,'endiX',0,'endiY',0,'midiX',0,'midiY',0,'count',0,'a',0,'v',0.5,'sing',0,'cosg',0);
h(i) = plot(car(i,3),car(i,4),'or', 'MarkerSize', 8,'MarkerFaceColor','r');%调节 marker-
size 后的数值调节圆的大小
end
%货架结构体，包含货架货架坐标等信息
for j=1:HUOJIANUM
    uuhuojia(huojia(j,1))=struct('state',0,'num',huojia(j,1),'x',huojia(j,2),'y',huojia(j,3));
     plot(huojia(j,2),huojia(j,3),'square', 'MarkerSize', 20,'MarkerEdgeColor','k')% 画
```

```
货架
end
%订单结构体，包含订单状态，货物数量，剩余未处理货物数量信息
 norder=struct('state',2,'num',0,'count',0);
a1=zeros();%a(1,2)=货架编号
b=zeros();
carst=zeros(1,CARNUM);%数组用于存放机器人状态
fd = zeros();
ms = zeros();
park = [1;0];
mu = 0.3;%
max = 10;%订单包含最大货物数量
order = zeros();
for i =1:CARNUM-1
    park = [park,[park(1,i)+interval;0]];%用于存放停车点的货架编号
    %第二行用于存放车位是否空闲的状态        1：不空闲      0：空闲
end
mm = 1;%用于记录生成订单个数
p = 1-exp(-mu);%订单到达概率
for v=1:200%主程序循环
%判断是否订单到达
nn = 1;
a = rand;
if(a<=p)
R=1;
bb=randi(max);
M=randi(HUOJIANUM,1,bb);
for i=1:bb
order(mm,nn)=M(1,i);
nn=nn+1;
```

```
order(mm,nn)=0;
nn=nn+1;
end
order(mm,2*max+2)=bb;
mm=mm+1;
else
R=0;
end
if(R == 1)%有订单来到
    norder(mm-1)=struct('state',0,'num',order(mm-1,2*max+2),'count',0);
    norder(mm)=struct('state',2,'num',0,'count',0);
end
%订单分配
for i=1:CARNUM
     carst(1,i)=ncar1(i).state;
 end
if(min(carst)==0  && norder(m).state~=2)%只有存在空闲车辆且当前订单未完成分配时，才进行订单分配
    if(norder(m).state==0)
    [a1,b]=orderdistribution(m,order,ncar1);  %寻找最终打包货架和对应运输机器人并返回
    norder(m).state=1;    %订单正在处理阶段
    norder(m).count=order(m,2*max+2)-1;
    end
    %订单中货物状态：初始 0 ；已分配 1 ；未分配 2
    for i=1:2:2*order(m,2*max+2)%步长改为 2
        c=b((i+1)/2,2);
        if(order(m,i+1)==2)
            [num,ind]=min(carst);
            c = ind;
```

```
            order(m,i+1)=0;
        end
            if(ORDERNUM~=0  && order(m,2*max+2)==1)%给空闲的车分配任务，订单里只含有一个商品
                ncar1(c).state=0;      %车空闲
                norder(m).state=2;     %订单完成
                m=m+1;    %订单完成
            elseif(ORDERNUM~=0  && order(m,i)==a1(1,2)  && order(m,i+1)~=1)%该货架是最终打包货架，则跳过
                order(m,i+1)= 1;
                n=n+2;
                 if(n==2*order(m,2*max+2)+1)    %若最终打包货架是最后一个货物所在货架
                    m=m+1;    %订单号加一
                    n=1;    %从第一个货物开始
                    break;
                end
            elseif(ORDERNUM~=0&&ncar1(c).state==0&&order(m,i)~=a1(1,2)&& order(m,i+1)~=1)    %给空闲的车分配任务，订单里含有多个商品，处理的货物数小于订单内货物的总数
                    ncar1(c).state=1;    %机器人处于取货阶段
                    ncar1(c).count=m;      %正在处理第 m 个订单
                    order(m,i+1)= 1;
                    ncar1(c).begiX=ncar1(c).x;    %车的起点
                    ncar1(c).begiY=ncar1(c).y;
                    ncar1(c).endiX=uuhuojia(a1(1,2)).x; %货物的最终目的地
                    ncar1(c).endiY=uuhuojia(a1(1,2)).y;
fd(ncar1(c).count)=plot(ncar1(c).endiX,ncar1(c).endiY,'pentagram','MarkerSize',8,'MarkerEdgeColor','c','MarkerFaceColor','c');  %确定最优目的货架的位置标记成亮蓝
```

色

ncar1(c).midiX=uuhuojia(order(m,i)).x;　%取货货架的位置

ncar1(c).midiY=uuhuojia(order(m,i)).y;

ms(c)=plot(ncar1(c).midiX,ncar1(c).midiY,'pentagram','MarkerSize',8,'MarkerEdgeColor','b','MarkerFaceColor','b');　%确定中间取货货架的位置并标记成蓝色

lm(c) = plot([ncar1(c).x,ncar1(c).midiX],[ncar1(c).y,ncar1(c).midiY]);

n=n+2;

if(n==2*order(m,2*max+2)+1)　%当订单内货物分配完成后

m=m+1;　%订单号加一

n=1;　%从第一个货物开始

break;

end

elseif(ORDERNUM~=0 && order(m,i+1)==1)%货物已分配

elseif(ORDERNUM~=0 && ncar1(c).state~=0 && order(m,i+1)==0)%货物未分配且车辆不空闲

order(m,i+1)=2;%该货物优先度标记为 2

end

end

end

#寻找最终打包货架和对应运输机器人的函数

function [a,dcp]=orderdistribution(m,order,ncar)

global CARNUM max;

load huojia;

dcp=zeros(max,2);%用于存放车辆和货物之间的距离的数组

aa=zeros();

fxsum=zeros();

totalgx=zeros();

```
finalspot=zeros();
overall=zeros();%用于存储全局车辆距离货架远近的数组
largenum=1000000;
for i=1:2:2*order(m,2*max+2)
        fx=0;
    for j=1:2:2*order(m,2*max+2)
        if(i~=j)
        n1=order(m,i);%第 i 件货物所在货架编号
        n2=order(m,j);%第 j 件货物所在货架编号
        fx=fx+(huojia(n1,2)-huojia(n2,2))^2+(huojia(n1,3)-huojia(n2,3))^2;% 以 订
单中第 i 个物品为打包点的其余物品与该点路程之和的平方
        end
    end
    fxsum(1,(i+1)/2)=fx;%将计算路程放入数组
end

%找到运货的最优车辆
for i=1:2:2*order(m,2*max+2)
    for j=1:CARNUM
        if(ncar(j).state==0)
        n1=order(m,i);
        gx=(huojia(n1,2)-ncar(j).x)^2+(huojia(n1,3)-ncar(j).y)^2;
        aa(1,j)=gx;
        end
        if(ncar(j).state~=0)
            aa(1,j)=largenum;% 货 架 到 无 人 车 的 最 远 距 离 应 该 是
46^2+46^2=4232，用以除去不在空闲状态的车辆
        end
    end
```

```
    overall(1:CARNUM,(i+1)/2,1)=aa;%列存第 i 个货物距所有车的距离
    [mingx]=min(aa);%获得距第 i 个货物最近的车辆距离以及其编号
    totalgx(1,(i+1)/2)=mingx;%将每个货物最近的车辆距其距离存入数组
end

%计算综合最小值
for i=1:2:2*order(m,2*max+2)
    fx=0;
    for j=1:2:2*order(m,2*max+2)
        if(i~=j)
            fx=fx+totalgx(1,(j+1)/2);%除去货架 i 外所有货架与其最近车辆的距离相加
        end
    end
        finalspot(1,(i+1)/2)=fxsum(1,(i+1)/2)+fx;%将第 i 个货物所在货架与其他货架之间的距离和除货架 i 外所有货架与其最近车辆的距离之和相加放入数组
end
[totalmin,huojiaindex]=min(finalspot);%找到总路径和最小值和最终打包点货架
a=[totalmin,order(m,(huojiaindex*2-1))];%1*2

%处理车辆调用冲突
%将货物 i 与所有车辆的距离和对应车辆编号存入数组，并且升序排列
%overall(:,:,1)按列存放货物 i 与所有车辆的距离，overall(:,:,2)存放对应车辆编号
for i=1:2:2*order(m,2*max+2)
    temporary(:,1)  = overall(:,(i+1)/2,1);
    temporary(:,2)  = 1:CARNUM;
    temporary  = sortrows(temporary);%将矩阵按第一列从小到大排序
    overall(:,(i+1)/2,1)  = temporary(:,1);
    overall(:,(i+1)/2,2)  = temporary(:,2);
```

```
end
%找出拥有重复最优车辆的订单中货物并将其用次优车代替直至无重复车辆
%找出 overall(:,:,2)中第一行中重复数的位置，并且用下一行相应的数替换，直至第一行没有重复数字，并且将其对应的距离也替换掉
a1 = overall(:,huojiaindex,2);
overall(:,huojiaindex,2) = zeros();
x=overall(1,:,2);
for j=1:CARNUM
x_u = unique(x);
b=[];
for i=1:length(x_u)
    if(length(find(x==x_u(i)))~=1)
        c=find(x==x_u(i));
        b=[b,c(1,2:length(c))];
    end
end
if(isempty(b))
    break;
end
x(1,b)=overall(j,b,2);
overall(1,b,1) = overall(j,b,1);%将车编号对应的距离也覆盖过去
end
overall(1,:,2) = x;
overall(:,huojiaindex,2) = a1;
for i=1:2:2*order(m,2*max+2)
    dcp((i+1)/2,:)=[overall(1,(i+1)/2,1),overall(1,(i+1)/2,2)];
end
end
```

```
#传统模型 S 形算法
load car;
load huojia;
%  框图大小
set(gcf,'unit','centimeters','position',[10  5 20 13])
set(gca,'Position',[.15  .15 .8 .75]);
set(get(gca,'XLabel'),'FontSize',8);
% 初始化数据
global CARNUM t max;
t=1;
CARNUM=10;
HUOJIANUM=88;%货架数
axis equal
hold on
%创建机器人结构体
for i=1:CARNUM
ncar1(i)=struct('state',0,'move',0,'num',car(i,1),'x',car(i,2),'y',car(i,3),'begiX',0,'begiY',0
,'endiX',0,'endiY',0,'midiX',0,'midiY',0,'count',0,'a',0,'v',1,'sing',0,'cosg',0);
    h(i) = plot(car(i,2),car(i,3),'or', 'MarkerSize', 8,'MarkerFaceColor','r');% 调节
markersize 后的数值调节圆的大小
end
%创建货架结构体
for j=1:HUOJIANUM
   uuhuojia(huojia(j,1))=struct('state',0,'num',huojia(j,1),'x',huojia(j,2),'y',huojia(j,3));
    plot(huojia(j,2),huojia(j,3),'square', 'MarkerSize', 20,'MarkerEdgeColor','k')% 画
货架
end
%创建订单结构体
norder=struct('state',2,'num',0,'count',0);
n=ones(1,CARNUM);
```

```
m=1;%表示订单序号
carst=zeros(1,CARNUM);
mu = 0.3;
max = 10;
order = zeros();
v1 = 0;
itermpos =zeros(11,2,CARNUM);%用于记录一份订单中货物依次位置的数组
mm = 1;%用于记录生成订单个数
p = 1-exp(-mu);
  for v=1:200
%判断订单是否到达
nn = 1;
a = rand;
if(a<=p)
R=1;
bb=randi(max);
M=randi(HUOJIANUM,1,bb);
for i=1:bb
order(mm,nn)=M(1,i);
nn=nn+1;
order(mm,nn)=0;
nn=nn+1;
end
order(mm,2*max+2)=bb;
mm=mm+1;
else
R=0;
end
if(R == 1)
      norder(mm-1)=struct('state',0,'num',order(mm-1,2*max+2),'count',order(mm-
```

```
1,2*max+2));
      norder(mm)=struct('state',2,'num',0,'count',0);
end
%订单分配
 for i=1:CARNUM
      carst(1,i)=ncar1(i).state;
 end
 if(min(carst)==0  && norder(m).state  == 0)%有空闲车辆且有待分配订单
      itermpos1  = zeros();
      norder(m).state=1;
      c = find(carst==0,1);%找到第一辆空闲的车
      ncar1(c).state  = 1;
      ncar1(c).count  = m;
      %将订单中的货物所在货架横纵坐标存入数组
      for i=1:order(m,2*max+2)
         itermpos1(i,1)=uuhuojia(order(m,2*i-1)).x;
         itermpos1(i,2)=uuhuojia(order(m,2*i-1)).y;
      end
         itermpos2  = sortrows(itermpos1);%将矩阵按第一列从小到大排序,即按 x
坐标排序
         itermpos2  =[itermpos2;[3,1]];
         itermpos(1:size(itermpos2,1),:,c)  = itermpos2;
         m=m+1;
 end
 %路径规划
 %S-shape
 for d=1:CARNUM
 if(ncar1(d).state  == 1)
 if(ncar1(d).x>2.5  && ncar1(d).x<=3.2  && ncar1(d).y==1)%仍在入口处
      if itermpos(n(1,d),1,d)  == 3 &&(itermpos(n(1,d),2,d)  == 1)
```

```
            ncar1(d).endiX  = 3;
            ncar1(d).endiY  = 1;
        else
        if(itermpos(n(1,d),1,d)  == 1 || itermpos(n(1,d),1,d)  ==  9 ||itermpos(n(1,d),1,d)
== 17 ||itermpos(n(1,d),1,d)  == 25 ||itermpos(n(1,d),1,d)  == 33 ||itermpos(n(1,d),1,d)
== 41)
        ncar1(d).endiX  = itermpos(n(1,d),1,d)+2;
        else
        ncar1(d).endiX  = itermpos(n(1,d),1,d)-2;
        end
        ncar1(d).endiY  = itermpos(n(1,d),2,d);
        plot(itermpos(n(1,d),1,d),itermpos(n(1,d),2,d),'pentagram',          'MarkerSize',
8,'MarkerEdgeColor','c','MarkerFaceColor','c');
        end
 elseif    (ncar1(d).x==ncar1(d).endiX    &&    ncar1(d).y==ncar1(d).endiY    &&
ncar1(d).endiY  ~= 1)%到达目标货架
            norder(ncar1(d).count).count  = norder(ncar1(d).count).count-1;
            %set(fd(ncar1(d).count),'visible','off');
            plot(itermpos(n(1,d),1,d),itermpos(n(1,d),2,d),'square',
'MarkerSize',20,'MarkerEdgeColor','k','MarkerFaceColor','w')
            n(1,d)=n(1,d)+1;
        if(itermpos(n(1,d),1,d)  == 1 || itermpos(n(1,d),1,d)  ==  9 ||itermpos(n(1,d),1,d)
== 17 ||itermpos(n(1,d),1,d)  == 25 ||itermpos(n(1,d),1,d)  == 33 ||itermpos(n(1,d),1,d)
== 41)
        ncar1(d).endiX  = itermpos(n(1,d),1,d)+2;
        else
        ncar1(d).endiX  = itermpos(n(1,d),1,d)-2;
        end
        ncar1(d).endiY  = itermpos(n(1,d),2,d);
        if itermpos(n(1,d),2,d)~=1
```

```
        plot(itermpos(n(1,d),1,d),itermpos(n(1,d),2,d),'pentagram', 'MarkerSize',
8,'MarkerEdgeColor','c','MarkerFaceColor','c');
        end
  end
    ncar1 = sshapepath(ncar1,d,itermpos,n,h);
    if(norder(ncar1(d).count).count==0 && ncar1(d).x>2.5 && ncar1(d).x<3.5 &&
(ncar1(d).y==1))%如果机器人回到入口处
        ncar1(d).state = 0;%机器人空闲
        norder(ncar1(d).count).state=2;%订单完成
        n(1,d)=1;
        ncar1(d).endiX = 0;
        ncar1(d).endiY = 0;
    end
  end
  end
  end
```

%S 形路径算法

```
function [ncar] = sshapepath(ncar,d,itermpos,n,h)
global t;
      if(ncar(d).x ~=ncar(d).endiX) %先沿 x 轴平移
          if((ncar(d).y<=26 && ncar(d).y>=18 || (ncar(d).y<=14 && ncar(d).y>=6))
&& (n(1,d)>1) && (itermpos(n(1,d),1,d)~=itermpos(n(1,d)-1,1,d)+6) || ncar(d).endiY
== 1)
                    %先运动到横向通道,离下一位置最近的
                    if(ncar(d).endiY == 1)
                      ncar(d).midiY = 1;
                    elseif(itermpos(n(1,d),2,d)<=13)%移动到最下面
                      ncar(d).midiY = 3;
```

```
                elseif(itermpos(n(1,d),2,d)>=35)%移动到最上面
                  ncar(d).midiY  = 28;
                else%移动到中间
                  ncar(d).midiY  = 16;
                end
                  ncar(d).midiX  = ncar(d).x;
[ncar(d).x,ncar(d).y,ncar(d).v,ncar(d).sing,ncar(d).cosg]=pathmove(h(d),ncar(d).x,ncar
(d).y,ncar(d).midiX,ncar(d).midiY,ncar(d).a,ncar(d).v,t);
        elseif  ncar(d).y==1  && ncar(d).endiX>=9
[ncar(d).x,ncar(d).y,ncar(d).v,ncar(d).sing,ncar(d).cosg]=pathmove(h(d),ncar(d).x,ncar
(d).y,ncar(d).x,ncar(d).y+2,4,ncar(d).v,t);
        else
              ncar(d).midiX  = ncar(d).endiX;
              ncar(d).midiY  = ncar(d).y;
[ncar(d).x,ncar(d).y,ncar(d).v,ncar(d).sing,ncar(d).cosg]=pathmove(h(d),ncar(d).x,ncar
(d).y,ncar(d).midiX,ncar(d).midiY,ncar(d).a,ncar(d).v,t);%平移
        end
    else%沿 y 轴运动
[ncar(d).x,ncar(d).y,ncar(d).v,ncar(d).sing,ncar(d).cosg]=pathmove(h(d),ncar(d).x,ncar
(d).y,ncar(d).endiX,ncar(d).endiY,ncar(d).a,ncar(d).v,t);
    end
end
```

致谢

四年时光，有如白驹过隙，转瞬即逝。回忆起刚踏入大学校门时懵懂的自己，如今已快要和相伴四年的母校告别了。回首这段时光，有欢笑，有泪水，有成长，也有挫败，还有陪伴我的同学和老师，在即将毕业之际，我希望对他们献上深深的感谢。

首先，我要感谢我的母校浙江工业大学。她是我四年以来成长的地方，承载了我的太多回忆。让我体验了人生之中最最不可或缺的一段宝贵经历。

其次，我要感谢我的指导老师赵云波教授。赵老师治学严谨，和蔼可亲，对学生们谆谆教诲，循循善诱，是学生们人生路上的一盏指明灯。在赵老师的指导下，我才能如此顺利的完成毕业设计的工作。在毕业设计中，每当我遇到困难，向他请教，他总是细心地为我解答，并且为我提供思路。在每一次的交流中，我都感觉获益良多。

最后，我要感谢我的母亲，没有她的支持和资助，我就不可能有机会接受这么好的教育，提升自己的能力与见识。你一直是我人生路上前进的动力，也是我背后最强的保护盾，今后的日子，我会更加努力，不负期待。

时光总是飞快，四年的大学生活，我收获的不仅仅是表面的知识，还有可贵的朋友，可亲的老师，以及能力的提升。愿我能披荆斩棘，直挂云帆济沧海！