

分 类 号 _____

密 级 _____

U D C _____

单位代码 10151

集装箱码头进口箱堆存优化问题研究

赵 成

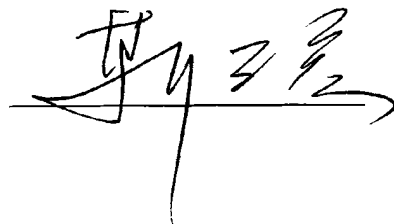
指 导 教 师 计明军 职 称 教授

学位授予单位 大连海事大学

申请学位级别 工程硕士 学科（专业） 物流工程

论文完成日期 2014 年 5 月 答辩日期 2014 年 6 月

答辩委员会主席





**Optimization for Allocation Problem of Import Container in
Container Terminal**

**A thesis Submitted to
Dalian Maritime University**

**In partial fulfillment of the requirements for the degree of
Master of Engineering**

**by
Zhao Cheng
(Logistics Engineering)**

Thesis Supervisor: Professor Ji Mingjun

May 2014

大连海事大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：本论文是在导师的指导下，独立进行研究工作所取得的成果，撰写成硕士学位论文“集装箱码头进口箱堆存优化问题研究”。除论文中已经注明引用的内容外，对论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本论文中不包含任何未加明确注明的其他个人或集体已经公开发表或未公开发表的成果。本声明的法律责任由本人承担。

学位论文作者签名：赵成

学位论文版权使用授权书

本学位论文作者及指导教师完全了解大连海事大学有关保留、使用研究生学位论文的规定，即：大连海事大学有权保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连海事大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。同意将本学位论文收录到《中国优秀博硕士学位论文全文数据库》（中国学术期刊（光盘版）电子杂志社）、《中国学位论文全文数据库》（中国科学技术信息研究所）等数据库中，并以电子出版物形式出版发行和提供信息服务。保密的论文在解密后遵守此规定。

本学位论文属于： 保 密 ☐ 在____年解密后适用本授权书。

不保密 ☒ （请在以上方框内打“√”）

论文作者签名：赵成 导师签名：张

日期：2014 年 6 月 19 日

摘要

集装箱码头作为集装箱运输的重要环节，其运营效率直接影响到港口的核心竞争力，如何提高集装箱码头的作业效率已成为当前集装箱码头研究所关注的焦点。吞吐量增长的同时也带来了码头管理的难度和复杂程度的增加，面对不断增加的箱量，考虑到有限的堆场空间资源，有效的堆存策略能提高码头的作业效率，不合理的堆存策略将会增加堆场的翻倒箱作业，降低码头的作业效率。合理安排堆场计划，不仅能降低翻箱量，提高码头运转效率，而且还能最大限度地提高空间使用率以及货主服务满意度，对码头方以及货主都有很大的意义。

本文结合集装箱码头实际作业情况，研究船舶卸载过程的集装箱堆存计划。文中根据进口箱预约提箱时间以及卸船计划，考虑到当前的箱位分配方案影响着后续提箱时的翻箱量，以提箱过程中总的翻箱量最小为目标制定了一个最优的进口箱箱位分配方案；同时为了减少二次翻箱，给出了一定的堆场翻箱策略来指导翻箱。以此为基础，在满足集装箱最大堆存时间、码头机械作业能力以及堆场面积等约束条件下，建立堆场进口箱堆存优化模型。利用遗传算法设计求解该堆存优化模型，最后通过分析不同规模算例的优化结果来验证模型及求解算法的合理性和有效性。

研究表明：本文给出的堆场堆存方案可以在兼顾卸船计划以及提箱时间的基础上有效减少翻箱，按照给出的翻箱方案进行翻箱操作能够减少二次翻箱，而且可以快速地为操作人员制定出理想的操作方案，提高空间利用率，增强港口竞争力。

关键词：进口箱；卸船计划；箱位分配；遗传算法

ABSTRACT

Container terminal is a very important part in container transportation, the operation efficiency directly affects the core competitiveness of the port, how to improve the operation efficiency has become the focused area of current research. The growth of container throughput makes their managements more difficult and complicated. Faced to the increasing containers and the limited storage area, reasonable allocation plan can improve work efficiency, however, the unreasonable allocation plan can raise the shift rate and reduce work efficiency. Reasonable stacking plan can not only reduce the turnover times, as well as improve the terminal operation efficiency, but also maximize the space utilization and the owner's service satisfaction which has great significance for port and shipper.

Based on the actual operating condition of ports, this paper is to do a research on container allocation plan in the process of unloading. According to the booking time to extract the container and the unloading plan, considering the influence the current storage location made on subsequent container, this paper is to determine the best allocation plan for import container, the plan is to minimize the turnover times during the pickup operation. In order to reduce the number of relocations, a decision rule is proposed. On the basis of the analysis, considering the constraint of storage time, the largest container port machinery operating capacity as well as the yard area, we establish an optimization model of import container stacking in the terminal yard. An improved genetic algorithm is designed to solve the storage optimization model and different experiments are used to validate the proposed model and the algorithm are effective and reasonable.

The results show that the proposed allocation plan can reduce the number of turnover times, According to the proposed turnover plan, container terminal can reduce the number of relocations, in the meanwhile it can quickly develop an ideal arrangement which can improve the space utilization and enhance the competitiveness of the port.

**Key Words: Import Container; Unloading Plan; Container Allocation;
Genetic Algorithm**

目 录

第1章 绪论	1
1.1 研究背景及意义	1
1.1.1 研究背景	1
1.1.2 研究意义	2
1.2 国内外研究现状	2
1.3 本文研究内容及组织结构	6
1.4 本章小结	8
第2章 集装箱堆场进口箱提箱翻箱问题分析	9
2.1 集装箱码头堆场概述	9
2.1.1 集装箱码头的布局及其设施	9
2.1.2 集装箱堆场的主要业务内容	11
2.1.3 进口箱卸船及提箱作业流程	13
2.2 集装箱在堆场中的箱位编码及堆存规则	15
2.2.1 集装箱在堆场中的堆存状态及箱位表示	15
2.2.2 集装箱在堆场中的堆存规则	16
2.3 进口箱提箱时翻箱问题分析	18
2.3.1 集装箱翻箱问题的分类	18
2.3.2 进口箱翻箱原因分析	20
2.3.3 降低进口箱提箱时翻箱的方法	20
2.4 本章小结	21
第3章 集装箱堆场卸船时箱位分配问题优化模型	22
3.1 进口箱堆存优化问题描述	22
3.1.1 问题描述	22
3.1.2 问题分析	23
3.2 进口箱堆存优化模型	27
3.2.1 模型假设	27
3.2.2 建立数学模型	27
3.3 本章小结	29
第4章 进口箱箱位分配问题算法设计	30
4.1 遗传算法概述	30
4.1.1 基本思想	30
4.1.2 基本步骤	30

4.2 进口箱堆存优化模型算法设计	32
4.2.1 染色体的编码及初始解的产生	32
4.2.2 适应度函数的设计	34
4.2.3 遗传算子的确定	35
4.3 本章小结	36
第 5 章 算例分析	37
5.1 算例背景	37
5.2 算例求解	37
5.3 不同规模下的算例求解分析	41
5.4 本章小结	43
第 6 章 总结与展望	44
6.1 全文总结	44
6.2 未来展望	44
参考文献	46
附录 A 100 个进口箱箱位分配主程序代码	51
附录 B 翻箱量子函数代码	56
致 谢	62

第 1 章 绪论

1.1 研究背景及意义

1.1.1 研究背景

日益深入的经济全球化带动了国际贸易的蓬勃发展,集装箱运输以其效率高、成本低、装载量大的优点契合了货物运输全球化和高效化的需求,直接推动了航运业的繁荣发展。各国为加速其运输现代化进程,均致力于集装箱运输的普及和发展。我国的集装箱运输始于 1973 年,经过了几十年发展,如今已跻身于航运大国之列。欧美等发达国家集装箱化程度较高,集装箱运输将是发展中国家和地区运输发展的趋势,在这一机遇面前,我国港口集装箱的吞吐量必将继续保持高速增长,同时这对我国港口的管理提出挑战。

由于货运量的飞速增长,码头堆场面积显得相对不足。集装箱堆场在集装箱综合运输体系中占有很大地位,作为码头堆存集装箱的重要场地,几乎所有的进出口作业都涉及到集装箱堆场。堆场作为港口物流系统中的重要节点,其作业效率直接影响到整个物流系统。堆场管理主要包括两方面,一方面是集装箱堆存时的箱位分配,另一方面是堆场相关作业机械的调度。从投资收益的角度讲,港口在开展集装箱运输初始,需要投入很多资金来完善港口的基础设施使其达到港口运营的条件,在日常运营中应充分合理地利用港口机械设施以及场地空间,保证港口的收益最大化。在堆场面积有限的前提下,为了充分利用堆场的堆存空间,集装箱码头现在普遍采用多层堆存策略,但是随着集装箱堆存高度的不断增加,提箱时的翻箱量也会不可避免的增加,这势必会降低堆场作业效率,增加堆场机械损耗。由此可见,在当前港口的布局规划、设备设施的配置均已固定时,翻箱量的多少已成为影响码头装卸效率和运营成本的关键因素。

从物理布局上来讲,堆场在码头中占有最大的面积,而堆场为进出口集装箱分配的箱位的合理性直接影响到堆场的作业效率。合理的堆存方案不仅能降低翻箱量,加快集装箱的周转速度,而且还能充分利用堆场面积和码头通过能力,降低码头运营支出。因此,高效的堆场管理是提高堆场效益、解决港口低效率问题的关键因素。由于前方堆场的主要作用在于加速船舶装卸集装箱,功能单一且

很难改变，后方堆场面积最大、功能较多，对后方堆场功能的优化越来越受到人们的关注。只有采用科学的堆存策略对堆场进行有效的管理，才能进一步提高堆场利用率，在降低运营成本的同时带来更大的收益。然而现有的堆存策略是将同一货主或同一提单的进口箱放在同一贝内，不同提单或不同货主的集装箱一般放在不同的贝上。虽然这种堆存策略在一定程度上能使货主在提箱时不发生翻箱，但是其浪费了堆场空间资源以及增加了卸船作业过程中的机械磨损和人工时间，这也在一定程度上导致了目前我国集装箱堆场的利用率普遍偏低的现象。因此，进口箱卸船时在集装箱堆场的箱位分配问题就显得非常重要。

1.1.2 研究意义

随着近年来世界集装箱吞吐量的不断增长，现有堆场的各项性能指标已显得相对落后，另外由于集装箱码头装卸企业的生产具有不间断性、协作性、不对称性等特点，所以为了保证港口码头的高吞吐量和快速高效的服务，堆场管理就显得尤为重要。箱位分配作为堆场管理的重要内容，采用什么样的分配方案以方便合理地安排进口箱在堆场中堆存，对于提高堆场利用率和提箱作业的效率有很大意义。本文仅从优化箱位分配的方面对堆场管理展开研究，以一次卸船的进口箱为决策对象，在堆场分堆的操作规则下基于已经确定的卸船顺序和提箱顺序，为每一个按照卸船顺序到达堆场的进口箱分配最佳箱位。其目标是货主提箱时，后方堆场在操作过程中产生的翻箱量最少。对码头而言，作为集装箱的中转地，降低翻箱率能够提高堆场的作业效率，减少堆场作业时间，进而增强港口的综合竞争力；对于客户而言，降低翻箱率能够使其在提箱时减少等待时间，在一定程度上提高了堆场的服务满意度，同时能够提高货物的周转速率，抢占先机，促进企业发展。因此，港方与货主都迫切希望降低堆场的翻箱率，集装箱码头进口箱堆存优化问题的研究在现实中具有重大意义。

1.2 国内外研究现状

集装箱堆场作为集装箱运输体系的重要组成部分，在航运事业中有着举足轻重的地位。随着集装箱运输在全球范围内得到认可，各国都大力发展集装箱运输并把其发展程度作为运输现代化的标志。堆场管理是港口物流链中重要而且难度

较大的一环,为了使堆场能容纳更多的集装箱以及更好更快的装卸船,必须要有一套切实可行的堆存计划。堆场堆存计划的评价指标通常是提取集装箱时堆场中产生的总的翻箱量,目前许多堆场都面临提箱时翻箱量过高的问题,这不但浪费了码头的资源,同时也降低了货主的满意度。为了减少堆场翻箱量,企业和各国航运界学者都致力于此类问题的研究。其中涉及到的相关内容有堆场进口箱翻箱量的计算、堆场中箱区的分配、具体箱位的确定问题以及翻箱时落箱位置的选择问题。

(1) 对堆场翻箱问题的研究开始于对翻箱量的估计。Kim^[1]早在集装箱研究的初期就对翻箱量的估计提出了自己的方法,对于不同布局和规模的堆场,根据其在提取过程中对平均翻箱量的影响,建立了非线性回归方程,并根据模型特点对其进行了简化求解。Kim^[2]忽略客户提箱时间对翻箱量的影响,假设堆场集装箱的堆存状态和提箱次序是确定的,按照提箱顺序依次提走贝中的集装箱,并利用分支定界法和启发式算法分别计算提走贝中所有集装箱所需的翻箱量。集装箱堆场中分别包含了进口箱作业和出口箱作业,在对进口箱进行提箱操作时,影响货主提箱的因素比较多,提箱时间的浮动性比较大,使得码头提箱时产生较多的翻箱量。减少进口箱翻箱量的主要途径是① 卸船时优化进场箱位分配,减少首次翻箱;② 翻箱时优选落箱位置,减少二次翻箱。相对于减少二次翻箱,优化进口箱的箱位分配更直接有效,是减少翻箱量的首选。现有研究通常将堆场集装箱箱位分配问题分两步解决:首先分配箱区,其次对箱区中具体的箱位进行确定。

(2) 在箱区分配问题的研究上,Roll等^[3]将堆场箱区的分配问题类比为物流仓库中货物摆放位置的确定问题,得出了集装箱在堆存时做好分类有利于减少翻箱时的工作量。Taleb-Ibrahimi等^[4]分析了集装箱在堆场的位置分配问题,提出可以临时提供一个缓冲区,专门堆放不按顺序到达的集装箱,通过比较分析进出口箱在不同放置规则下所需的堆场面积,得出了相应的堆存工作量。文中得到以下结论:在无临时缓冲区的情况下,虽然在一定程度上减少了堆场的操作,但是堆场空间利用率不高;在有临时缓冲区的情况下,虽然堆场空间利用充分,但翻箱不可避免。研究也表明,无论在上述哪种放置策略下都不能避免翻箱。Zhang等^[5]提出一种混堆策略:采用整数规划的方法把对每艘船要装卸的集装箱分配到各个箱区中

去,大大减少了船舶的滞港时间。王斌^[6]在求解混堆问题时,将集装箱堆场堆存问题分为两个过程:第一步平衡各箱区内总的进出口箱量,第二步考虑到货主提取集装箱时的集卡行驶距离,在进行目标优化时减少集卡行驶距离。刘艳^[7]将具有某些特性的集装箱进行分组,在第二阶段的目标中增加了使同组箱所占箱区数量最少的约束,加速了船舶装卸效率。对于此问题,陶经辉等^[8]采用混堆模式在考虑到总工作量平衡和箱组平衡的基础上用滚动规划法研究了区段分配问题,算例分析结果证明该模型和算法在很大程度上优于堆场现有的策略。Zhang^[9-10]通过分析货主提箱时间,以平均分配各箱区的箱量和减少集卡提箱时总的行驶距离为目标,建立了堆场区段两阶段分配模型,并根据模型特点将目标函数由非线性转换为线性,提高了求解效率。岳文英等^[11]研究建立了前、后方堆场分开堆存进、出口箱的混合整数模型,该模型以最小化集卡行驶距离和平衡各箱区作业量为目标,利用lingo计算求解,所得优化结果表明该模型在实现集卡行走距离最短的前提下,同时也保证了龙门吊在箱区间作业量的总体平衡。

(3) 在具体箱位分配问题的研究上,金海龙^[12]对紧凑型集装箱码头进口重箱堆存规则进行了分析,得出结论:① 降低集装箱的堆存高度,可以明显减少进口箱的翻箱量,并有助于节约吊箱成本;② 将同一货主的进口箱在同贝位内摆放,也在一定程度上减少了翻箱量。周留井^[13]基于高效利用堆场的存储空间和减少翻箱量的考虑,采用混合堆存策略,利用遗传算法为每一个动态到达后方堆场的进口箱分配贝位,使翻箱量最少。杨超等^[14]通过对目前港口送箱时间窗的分析,发现在某一段时间内会发生拥挤,根据这一现象建立数学模型,旨在平衡每天送箱作业量的差异以及集装箱装船前在堆场的总逗留时间,运用遗传算法求解。数值实验表明,该模型能减轻作业量不平衡程度,提高堆场服务满意度。魏航^[15]构建了码头方的补助金与车辆方的成本函数,根据收益最大原则确定了进口箱取箱时间窗,通过对不同模式的对比,进行灵敏度分析。Kim等^[1]考虑到集装箱在配载时重箱压轻箱这种堆存策略,以最小化装船时的翻箱量为优化目标,利用动态规划和决策树方法为到来的集装箱分配位置。孙俊清^[16]讨论了集港时出口集装箱在堆场中的位置分配方案,考虑装船时间以及箱重因素,建立了动态规划模型。鉴于其时间复杂度,采用遗传算法进行求解,通过实例表明利用遗传算法可以更快找到

最佳箱位。郝聚民^[17]向我们阐述了图搜索技术和模式识别理论的有关知识,并基于以上两种方法建立了在不确定条件下混堆堆场中贝位分配优化模型。模拟结果表明,通过本文所建的模型能够做到出口箱在装船时不产生翻箱操作。陈庆伟^[18]根据混堆工艺,以翻箱量最小为约束,为每一个动态到达的集装箱分配合适的贝位,通过实例说明优化结果能够使未来的翻箱量较小。谢尘^[19]针对出口箱在堆场内选位问题,提出了选位优化策略,采用启发式算法和分布式遗传算法相结合的混合算法求解选位模型。仿真案例表明,该模型和算法在装船过程中能够降低船舶上的压箱数,同时对减少集装箱船的在港时间也有一定的促进作用。周鹏飞^[20]探讨了不确定环境下的进出口箱箱位优选问题,降低了堆场一次翻箱率。

(4) 对于进口箱翻箱时落箱位置的选择问题,国外学者的研究开始的较早,并且取得了一定的成果。基于减少货主提箱时的翻箱量,提高服务满意度的考虑, Kim and Hong^[2]在已知提箱顺序的基础上探讨了提箱过程中集装箱翻箱优化的分支定界法,并利用OH算法优选翻箱时落箱位置。徐亚等^[21]深入分析堆场翻箱时集装箱的落箱位置,在此基础上提出了改进的IH算法,仿真实验表明改进的IH算法优于OH算法。Lee and Hsu^[22]以最小化装船前预翻箱过程中的翻箱量为目标,提出了相应的整数规划模型,并利用启发式方法对落箱位置进行优化。Lee等^[23]利用领域搜索启发式方法优化了堆场预翻箱方案。Nang^[24]把出口箱以装船过程中产生翻箱量期望最小作为优化目标,同时针对进口箱以每提一个箱所需翻箱量的平均值最小为优化目标,分别建立了翻箱落位优选模型来选择集装箱的较优落箱位置。易正俊^[25]将落箱时翻箱位置的寻优问题转化为求解最短路径问题,在算法上采用了脉冲耦合神经网络,经验证脉冲耦合神经网络能够较快解决问题。王洪雪^[26]提出了一种把作业人员作为主体、集装箱作为客体、集装箱被取走时间作为主体决策的想法,并且设计了一种新的动态判断式优化算法,通过一个优化模型以及仿真结果得出了最优箱位分配的原则。董琳^[27]针对堆场中的实际翻箱问题,利用图论知识构建堆场预翻箱模型,并用加以限制的广度搜索算法计算出能够实现箱位堆放要求的翻箱的最小路径。易正俊^[28]采用博弈论的知识,把码头和客户看成局中人,提箱时间作为局中人的策略,通过建立码头局中人的效用函数,把寻求倒出箱的落箱最优位置转化成为计算效用函数的最大值问题。仿真实验表明,该方法

与目前国内外文献的同类优化算法相比较,效果明显。李斌^[29]基于动态规划算法制定贝内翻箱优化方案,并给出了堆场一些切实可行的翻箱操作方法来指导实际堆场中的翻箱。

通过以上分析我们可以看出堆场箱位分配问题已引起了国内外同行的广泛关注。由于以前信息化普及率较低,堆场不能及时获取集装箱的提箱时间,使得关于进口箱堆存的研究比较困难。以往对集装箱堆场箱位分配的研究主要集中在出口箱问题上,针对进口箱的研究往往较少,而且很大一部分研究只是论证了算法的可行性,并没有很好的与堆场中的实际作业相结合,给堆场提出具有指导性的建议,把进口箱的卸船与堆场堆存结合起来研究的就更少。本论文从全局考虑,通过分析堆场相关作业流程以及现有约束,建立了进口箱后方堆场堆存优化模型,并利用遗传算法来确定箱位分配方案,验证了模型的合理性与算法的有效性,在针对减少二次翻箱的问题上也给出了一些翻箱策略来指导翻箱。

1.3 本文研究内容及组织结构

本文主要是通过分析集装箱从开始卸船到被客户提走这一系列的过程,了解到集装箱在卸载过程中的箱位分配方案与集装箱的提箱顺序共同决定了进口箱提箱时的翻箱量。在满足堆场作业能力的条件下,结合客户的预约提箱时间,探讨了基于提箱时间的集装箱翻箱量的求解,并在满足卸船计划的条件下建立了堆场进口箱箱位分配数学模型。通过对具体算例的求解得出了一个较好的堆存方案,使得提箱时堆场翻箱量较小;同时为了减少堆场的二次翻箱,文中针对翻箱时落箱位置的选择问题给出了一套完整的方案,来指导后方堆场管理员进行翻箱操作。文中最后通过设计合理高效的算法进行模型求解,为进口箱在集装箱堆场的实际堆存操作提供理论依据与技术参考。本文的主要研究内容如下:

- (1) 货主提箱时翻箱原因分析。通过对集装箱码头进口箱提箱时操作流程的分析,找到影响提箱时翻箱的因素,结合集装箱码头在长期堆场管理中积累的经验,给出了减少堆场翻箱量的措施。
- (2) 进口集装箱贝内具体箱位分配模型。结合进口箱进入堆场的顺序,以货主提箱时最少翻箱量为优化目标,建立了进口箱箱位分配数学模型。
- (3) 翻箱时落箱位置的确定。堆场中某些因素会导致堆场翻箱次数的增加,为了提

高堆场在应对这一问题时的能力，结合以往的研究成果，本文从二次翻箱的角度提出了一个翻箱时落箱位置的确定方案，并给出翻箱时具体落箱位置的确定规则。

(4) 利用遗传算法对模型进行优化求解。在具体运用遗传算法时，基于实际情况，本文选择十进制编码方式，在制定后续的交叉变异规则时，本文具体情况具体分析，设计了适合本问题的遗传算子。通过具体算例，验证了遗传算法对于求解这类问题的可行性。

本文的组织结构如下：

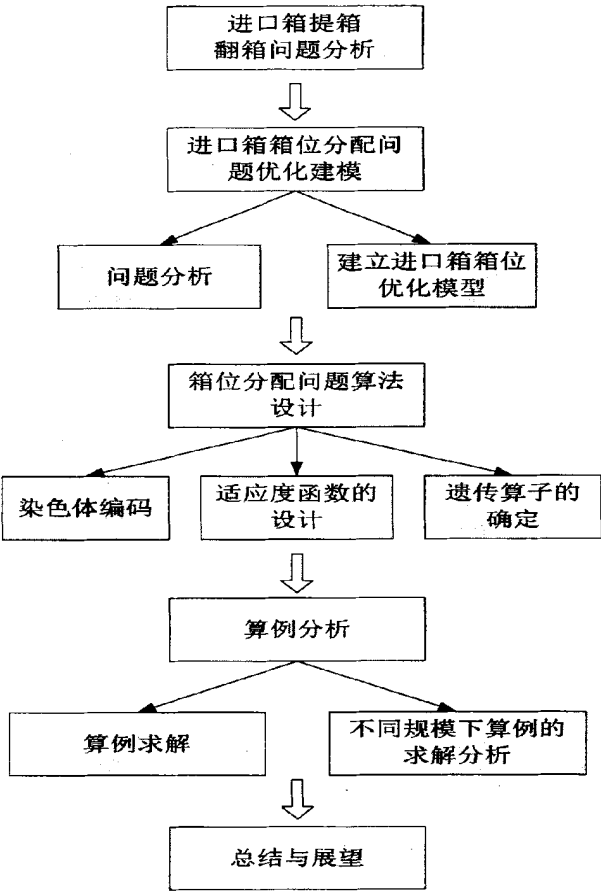


图 1.1 文章结构框架图

Fig. 1.1 Thesis structural frame graph

本文结合码头实际作业情况，重点研究进口箱从卸船到被提取的整个作业过程，从整体上设计一个进口箱后方堆场箱位分配方案以及翻箱策略，方便堆场管

理员根据卸船计划为进口箱分配合理的箱位，同时能够减少后续的翻箱量，提高港口作业效率。

1.4 本章小结

本章首先介绍了论文的研究背景和意义，然后从堆场进口箱翻箱量的计算、堆场中箱区的分配、具体箱位的确定问题以及翻箱时落箱位置的选择这几方面分别阐述了国内外的研究现状，通过分析各学者的研究，确定了本文的研究方向，最后概括了本论文的主要内容及技术路线。

第 2 章 集装箱堆场进口箱提箱翻箱问题分析

2.1 集装箱码头堆场概述

2.1.1 集装箱码头的布局及其设施

集装箱码头是供集装箱船停靠、装卸的作业场地，同时也是临时保管和向货主交接集装箱的场所。现代化的集装箱码头正在向高度机械化和生产大规模化的目标发展，其布局要以船舶作业为核心，将码头与船舶连接成一个有机整体，从而实现高效、有条不紊的连续作业。由于地理环境的不同，集装箱码头的布局也会有所不同，然而集装箱码头通常应具备以下基本设施：泊位、码头前沿、集装箱堆场、控制室、货运站、检查口、维修车间等。图 2.1 为某一集装箱码头布局示意图。

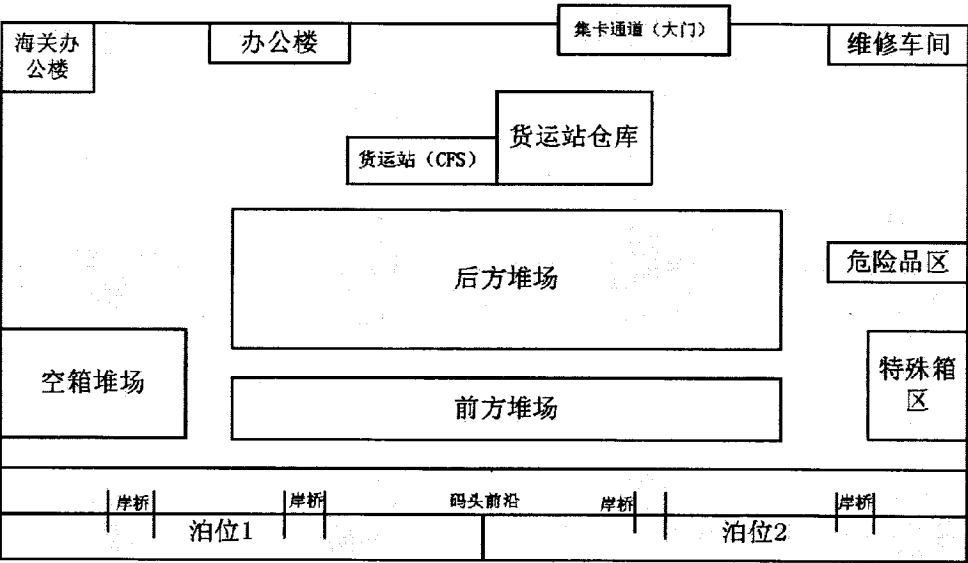


图 2.1 集装箱码头布局

Fig. 2.1 General layout of container terminal

(1) 泊位(Berth)

泊位是供集装箱船停靠和作业的场所。依据地理环境的不同在建造泊位时通常有三种方案：顺岸式、突堤式和栈桥式。通常集装箱码头采用顺岸式，其优点是建造成本低，从岸线到堆场距离较近，装卸船舶作业方便，同时对多个码头来

讲,还可以因装卸量的不同便于装卸桥在泊位间移动。

(2) 码头前沿(Frontier)

从地理位置上来看,码头前沿是指从泊位岸线的岸壁到堆场前沿之间的区域,其主要功能是为集装箱装卸桥以及牵引车提供工作空间。根据装卸机械配置的不同,其相应的码头前沿的面积也不同,一般来讲码头前沿的宽度必须满足所采用的装卸机械和搬运机械工作时对场地的要求。目前先进的集装箱码头前沿均配置岸边集装箱装卸桥,一些吞吐量较小的码头前沿则配置其他种类的高架起重机。

(3) 集装箱码头堆场(Container Terminal Yard)

集装箱堆场有些地方也称为场站,主要功能是临时堆存集装箱,出口箱在出口前会将集装箱送到堆场,直至装船完毕;船舶到港后,卸载下来的进口箱基于加速卸船的考虑会首先堆放到堆场中,直至在免费堆存期限内进口箱被提取完毕。按照功能的不同通常可分为前方堆场、后方堆场和空箱堆场。

① 前方堆场(Container Marshalling Yard)

前方堆场从物理布局上来看是位于码头前沿与后方堆场的部分,由于距离岸边比较近,更多的是堆存出口箱,其作用是:集装箱船靠泊前,按照配载要求将集装箱有计划的堆放;卸船时,为了加速船舶的卸载,通常将进口箱暂放在前方堆场。一个泊位的面积应至少为能堆放该泊位所能停靠最大船舶载箱量的两倍。

② 后方堆场(Container Yard)

后方堆场紧邻前方堆场,其占有最大的堆存面积,是码头堆放集装箱的主要部分,所有进口箱都放置在后方堆场,同时后方堆场也是“场到场”这种交接方式办理交接的场所。集装箱种类的多样性使堆场作业变得复杂,为了方便作业,后方堆场继续分为重箱箱区、空箱箱区、冷藏箱区及危险品箱区等。

③ 空箱堆场(Van Pool)

空箱在堆存时不用考虑集装箱的承重,可以更高效的利用堆存面积,且在进出口时都会涉及到空箱的收发,和空箱有关的业务更加繁忙。堆场通常专门划出一定的面积用于办理和空箱有关的业务,由于作业机械的限制,空箱堆场一般不办理重箱业务,它可以单独经营,也可以由集装箱装卸区在区外另设。

事实上在集装箱堆场上这几类堆场并没有严格的分界线,仅仅是地理位置上

的相对概念。在作业高峰期，前方堆场也可以临时放置进口箱。一般来讲出口箱为了方便装船要放在前方堆场，进口箱基于不阻碍出口箱装船的目的要放在后方堆场，空箱为了便于统一管理，要放在空箱堆场，特殊箱在堆存时有些需要接通电源，有些需要隔离，要放在相应类别的堆场中。

(4) 控制室(Operation Control)

控制室是集装箱码头的“心脏”，集装箱码头的各项操作都由控制室进行协调控制。随着集装箱信息化的不断推进，现代化的堆场多采用计算机生产作业系统进行统一管理，控制室成为码头各项信息的汇集处理中心，是集装箱码头最重要的部门。

(5) 货运站(Container Freight Station 简称: CFS)

作为集装箱运输关系参与方的组成部分，货运站在集装箱运输过程当中发挥的作用举足轻重。它主要负责拼箱货的拆箱、装箱以及拼箱货的交接作业，集装箱是在配、积载之后被送到集装箱堆场。货运站还接纳从堆场过来的进口货箱，并且负责对其实施拆箱操作、整理操作以及保管操作，直到最后将货物交给各个收货人。另外，货运站也可接受承运人的委托对货箱进行铅封以及场站收据的签发等多项业务。

(6) 检查口(Inspection Door)

检查口也被称为道口，也有的、称之为闸口或者大门，公路集装箱要想进出码头必须要经过检查口，同时它还是交接两方关于集装箱责任的划分点，另外在检查口也能处理一些有关集装箱进出口的业务操作。检查口一般情况设立于码头后方靠近大门的地方，根据处理业务的不同还可以细分为出场检查口与进场检查口两种。

(7) 维修车间(Maintenance Shop)

集装箱堆场中专用机械设备以及集装箱本身的维修和保养都是由维修车间负责完成的。由于集装箱码头对其所拥有的专用机械设备的性能要求比较高，从而来保证其装卸集装箱的作业效率，所以维修车间在平时就发挥着极其重要的作用。

2.1.2 集装箱堆场的主要业务内容

对于海运出口集装箱来讲，堆场的作用就是装船前一段时间内所有出口集装

箱在这里聚集，等到集装箱船到港后统一装船。相对应的对于进口箱来讲，堆场的作用是从船上卸下来的集装箱临时堆存在这里，在一定的时间内，进口箱被全部提走。总的来说，集装箱堆场的主要业务内容包括空箱的发放、进出口箱的临时堆存、集装箱的交接、装卸船计划、对特殊集装箱的处理等，同时还会对某些破损箱进行维修。以港口集装箱堆场为例，其主要业务如下：

(1) 集装箱堆存与保管

无论是进口箱还是出口箱，集装箱在进场后，场站要对集装箱内的货物负全责，同时货主应在堆场规定的时间内送提箱，堆场要为到来的集装箱制定箱位计划，提高堆场利用率。对于转运箱，应严格按照承运人的货物目的地制定合理的货物中转计划。

(2) 集装箱的交接

发货人和集装箱货运站将由其或其代理人负责装载的集装箱货物运至码头堆场时，设在码头堆场的闸口对进场的集装箱货物核对相应的单据。同时还应检查所提取的集装箱是否与堆场记录的集装箱相对应。

(3) 制定堆场作业计划并作业

堆场作业计划是对集装箱在堆场内进行装卸、搬运、贮存、保管的安排，这是为了经济、合理的使用码头堆场和有计划的进行集装箱装卸工作而制定。堆场作业计划的主要内容有：① 集港作业，② 进口作业，③ 场地及场地机械计划。

(4) 对特殊集装箱的处理

危险品箱分开放置：所有危险品进出口箱只能堆放在专门的箱区中，并且要按照不同货物相应的隔离要求进行分开堆放。

冷冻箱分开放置：冷冻箱由于要使用堆场中的电源，只能堆放在冷藏箱区中，如果集装箱既是冷冻箱又是危险品箱，就应把箱子放入冷冻箱区。

超限箱分开放置：超限箱、平板箱和框架箱应进超限箱区，超限箱一般只能堆1层高。

(5) 协调与处理好和船公司的业务关系

码头方与船公司分工协作，集装箱码头应保证：在集装箱船到来时安排合适的靠泊位以及足够的人力和机械，以保证装船速度。同时还要提供足够的场所，

保证集装箱作业及堆存空间。船公司应保证：向码头提供自己的船期，在船期有变化时，第一时间通知码头，让其做好相应的安排，同时要及时提供货物的进出口情况，以便堆场安排箱位计划。

2.1.3 进口箱卸船及提箱作业流程

随着集装箱运输事业的成熟与完善，各大港口在进口箱卸船以及提箱作业方面已经制定了一套完整的作业流程，目前各大港口进口箱卸船及提箱流程如图 2.2 所示。

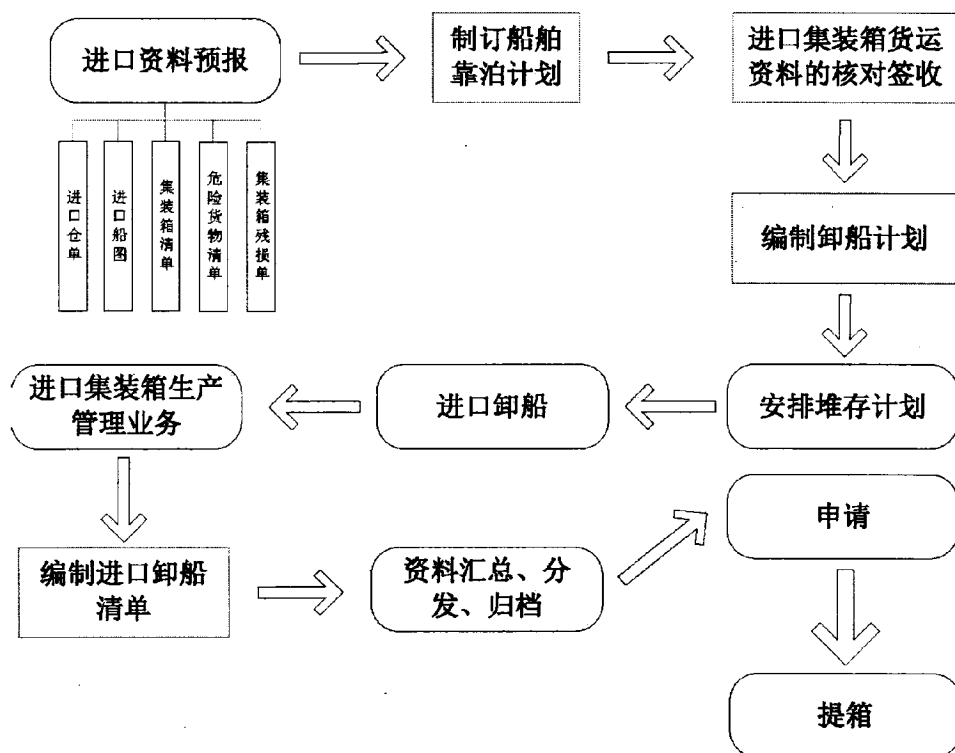


图 2.2 集装箱码头进口箱卸船及提箱业务流程

Fig. 2.2 Process of unloading the import containers and extracting the containers

进口箱卸船作业流程：

(1) 船代在船舶靠泊前一天将集装箱卸船清单和船图等船舶资料通过电脑发送到堆场单证室、配载、营业室，做好卸船前的准备工作。

(2) 堆场计划根据单证人员核对准确后的卸船清单和舱单，按进口重箱、进

口空箱、特种箱、中转箱合理安排堆场位置，积极与船控联系，满足船控的卸船要求。

(3) 配载员根据单证人员核对准确的卸船清单和舱单，结合卸船面图，通过电脑编制集装箱卸船顺序表，并打印出卸船顺序表，分别给理货、船控和装卸工。

(4) 中控室船控人员根据配载提供的卸船顺序表和堆场计划提供的场地位置，生成卸船面图和卸船分倍图。根据作业路数和场地的安排，布置好场地机械，准备卸船。

(5) 卸船时，船控发送卸船指令到船边外理手持，外理手持人员根据收到的指令，认真核对卸船箱箱号，仔细检查铅封和箱体的完整性，做好工原残记录，按指令将卸船箱的场地位置准确告诉集卡车司机，然后将集卡车车号和卸船指令进行确认。

(6) 场地机械根据收到的外理确认的指令，核对箱号和集卡车号，将卸船箱按照指令显示的位置进行标准堆码，然后进行卸船指令确认。

(7) 卸船完毕后，船控及时进行卸船完船确认。

进口箱提箱作业流程：

(1) 提货人提货前持加盖海关放行章的正本提单（所提货物如系木质包装还须加盖三检放行章，办理三检业务在三港池报检大楼）到营业大厅结算港口费。

(2) 放行员在港口费结清后在本公司提货凭证上加盖放提章交提货人。

(3) 提货人在提货前向市场部申报提货计划。

(4) 提货时，提货人车辆到港后需通知陆运调度派机械，然后持提货凭证与货物所在库场管理员核对货物。

(5) 提货完毕提货人和理货员共同在提货凭证上签字后将提货凭证交理货员，由理货员开内部门证。提货人持内部门证到门证组换取正式门证即可出港。如货物不能当天提清，提货人和理货员在提货凭证上签字后由提货人将提货凭证取回，并继续向市场部申报这批货物的提货计划。

码头通知该批集装箱的所有货主于一定时间内提走其集装箱。取箱车辆在规定期限内到港时间的无序性和不确定性是影响翻箱操作的主要因素，随着信息化

的发展,提箱预约方案的实施,未来码头有希望提前获取提箱时间。本文假定在卸船前码头已经获取了集装箱的提取时间,基于卸船计划,通过优化堆存方案减少提箱操作中的翻箱次数。

2.2 集装箱在堆场中的箱位编码及堆存规则

2.2.1 集装箱在堆场中的堆存状态及箱位表示

集装箱码头堆场由若干箱区组成,每个箱区又划分为若干个贝位,其中每个贝位又包括若干栈和层,这些要素就构成了箱位。堆场中每个箱位都可以通过其所在的箱区(block)、贝位(bay)、栈号(stack)和层号(tier)的编码来唯一确定,图2.3更直观描述了堆场中某一箱区的组成。

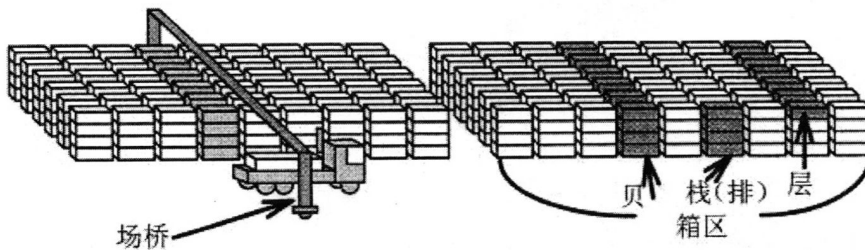


图 2.3 集装箱堆场某一箱区布局

Fig. 2.3 The layout of a block in container yard

为了方便集装箱的作业,堆场采用编码的方法对集装箱进行定位,这种方法能够确保每一个集装箱在堆场中的编码都是独一无二的。首先对箱区进行编码,箱区一般有两种编码方式:一种是由一位英文字母进行编码,另一种是一位英文字母和一位阿拉伯数字组合编码,英文字母在前,其中英文字母表示箱区对应码头的泊位号,阿拉伯数字表示从近海侧向后方堆场依次的顺序号,国内大多数堆场都是采用两位的编码方式。在对箱区中的贝位进行编码时,考虑到箱型的不同,奇数 01、03、05...表示 20ft 标准集装箱的贝位号,偶数 02、06、10...表示 40ft 或者 45ft 集装箱贝位号,贝位中的栈和层都用 1 位阿拉伯数字表示,栈数以及层数的设定取决于堆场相关作业机械的作业范围。比如“A20524”表示位于 A2 箱区第 5 贝第 2 栈第 4 层的标准集装箱。

文章重点探讨进口箱具体箱位的分配问题,假设一批待分配箱在堆场中已经

指定好堆放的箱区，这样就变成了箱区内贝、栈、层位置的确定问题。本文对堆场中的每一个箱位用三维坐标来表示，堆场中集装箱所在的贝、栈、层反映到堆场坐标系上如图2.4所示， X 轴坐标表示堆场中相对应贝位集合； Y 轴坐标表示堆场中相对应的栈集合； Z 轴坐标则表示集装箱所在的层，在这里以最接近地面的箱位作为第一层。

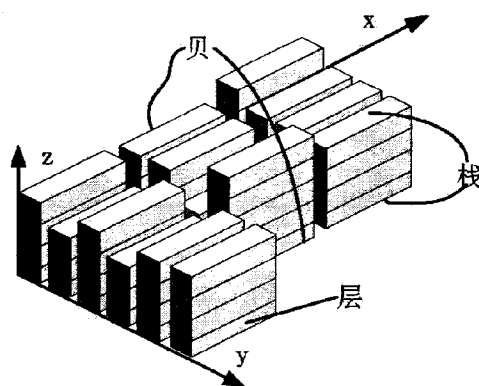


图 2.4 堆场中贝、栈、层与坐标轴的关系

Fig. 2.4 The relationship between the bay, the stack, the tier and the coordinate axes

2.2.2 集装箱在堆场中的堆存规则

堆场面积相对不足是目前各大港口普遍面临的问题，集装箱种类的多样化又增加了堆存问题的复杂性。很早码头方就已经意识到了这个问题，在逐渐的实践中，堆场已经找到了适合自己的堆存规则，然而这种规则更多的是出于经验考虑，效率并不高。无论是进口箱还是出口箱，在其入场堆存时都要考虑到对后续作业的影响。对于进口箱在放箱时通常有两种堆存策略：

(1) 进口箱全场混堆

在集装箱船靠泊后，待卸进口重箱在后方堆场安排箱位时，忽略船名航次的不同，将进口箱随机分配至有放箱位置的进口箱区各块。这种放箱规则要求同时开工的场地作业机械较多，机械利用率偏低，虽然加快了卸船速度，但导致后续提箱时轮胎吊大车移动频率和堆场翻箱率都比较高。

(2) 进口箱按船名航次堆存

同船名航次的集装箱按照自然箱箱型集中堆放于多个块中。这种放箱模式在灵活性上较混堆要差，降低了箱区利用率，但在一定程度上能够提高机械利用率减少提箱时的翻箱。

由于不同船名、航次或卸船日期进口的集装箱出场日期集中性较差，因此在同一贝位中尽量集中堆放出场日期相近的集装箱以降低轮胎吊大车移动频率。

根据堆场计划，在对出口箱进行箱位分配时大多遵循 PSCW 准则。当出口箱装船时，为保证船舶的稳性以及减少中途翻箱，出口箱在进场时会考虑重压轻、远压近的放置方案。目前，集装箱码头常用的出口箱堆存策略有 6 种，表 2.1 为这 6 种放箱方案的对比。

表 2.1 出口箱堆存规则

Tab. 2.1 Strategies of stacking rules of export containers in container terminal yard

堆存策略	堆存方案	分类标准
1	同尺寸、同船舶、同港口的集装箱堆存在同一贝，且贝内不同重量等级按重压轻堆放	尺寸、船名、卸货港、重量级
2	同尺寸、同船舶、同港口或相邻卸货港的集装箱堆放在同一贝，不同重量级按“重压轻”堆放	尺寸、船名、卸货港
3	同尺寸、同船舶的集装箱堆放在同一贝，不同卸货港按栈堆放	
4	同尺寸、同船舶的集装箱堆放在同一贝，不同卸货港“按远压近堆放”	尺寸、船名
5	同尺寸的集装箱堆放在同一贝，贝内不同船舶的集装箱按栈堆放	
6	同尺寸的集装箱堆放在同一贝，不同船舶的集装箱按“先作业压后作业”方式堆放	尺寸

上述 6 种堆存策略是按照堆场可计划空位由多到少进行排序的，策略 1 和策略 2 适用于堆场集装箱规模较小的情况，相对于此种情况下的集装箱数量，堆场的可计划空位较多。当堆场集装箱数量逐渐增多时，如果堆存时分类过细，堆场面积会严重不足，只能以提高翻箱率为代价来保证堆场堆存能力。出口箱集港时要综合考虑未来短时间内堆场的集装箱变化情况，根据出口计划为出口箱选择合适的堆存分类标准，减少装船时由于配载计划和堆存状态不一致造成的翻箱。

较为理想的堆存方案是不同货主的进口箱混放在一起，这样的做法能够保证在充分利用堆场空间资源的情况下，既提高工作效率，同时又减少了大车行走距离。但是，这样做在提箱时很可能出现大量的翻箱，翻箱操作不仅浪费了堆场的资源，而且降低了堆场的作业效率，在任何操作过程中都应该避免，这就对堆存策略的选择提出了很高的要求。

2.3 进口箱提箱时翻箱问题分析

2.3.1 集装箱翻箱问题的分类

箱位是集装箱在堆场上堆存的基本单位，集装箱在堆存时，通常是几个集装箱是垂直压在一起的，这种堆存方式就决定了只能对最上层的集装箱进行操作。由于堆场箱位安排的不合理或者货主提箱时间的临时变动，货主在提取集装箱时，如果目标箱并非处于堆场中某一栈的最上层，此时就需要将压在目标箱上方的集装箱移动到同一贝位的其它栈，这种把集装箱从一个栈移动到另一个栈的操作就被称作翻箱。图 2.5 为在堆场中某个贝位进行提箱操作时的示意图，箱子上面的数字代表了提箱时间段，数值越小表示越早被提取。提箱时假设第 6 栈中的 10 号箱为目标箱，由于其正上方的 15 号箱提箱时间比较晚，阻碍了该操作，在这里 15 号箱则被称为阻塞箱，那么就需要先将 15 号箱移动到其他栈再提取 10 号箱，这样对集装箱 15 的操作就称为一次翻箱操作。

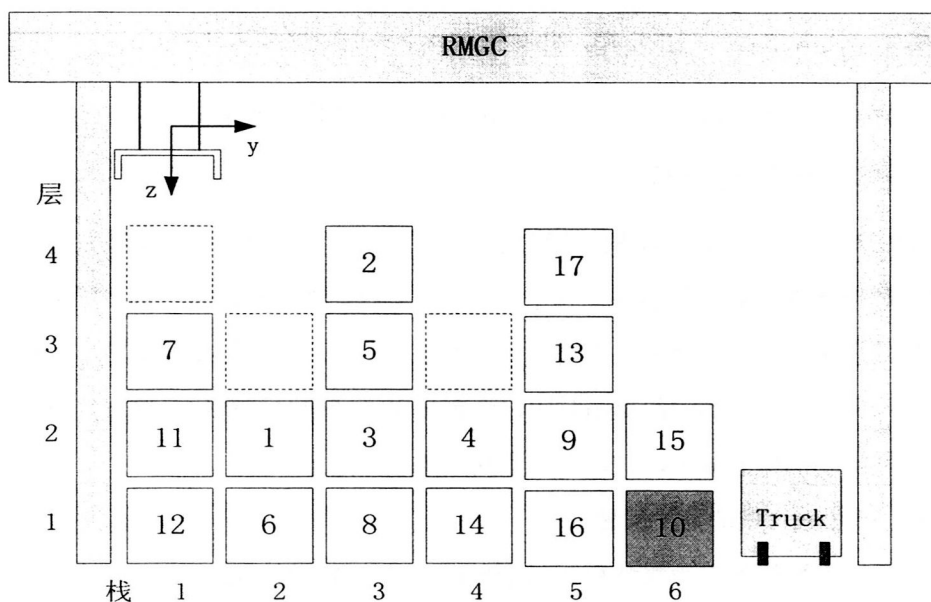


图 2.5 提箱操作示意图

Fig. 2.5 Yard pick-up operation

翻箱是一种无效作业行为，不产生经济收益，占用堆场资源，对港口服务效率有直接影响，因此，在港口实际操作中应该尽量避免翻箱操作。本文基于堆场中这一现实需要，针对堆场实际作业时出现的翻箱问题展开研究，目的是利用现有信息以翻箱量最小为目标进行进口箱箱位分配。

在集装箱堆场中翻箱主要分为移箱翻箱、装船翻箱和提箱翻箱这 3 种类型^[32]。

移箱翻箱通常受海关监管影响，根据我国法律的相关规定，集装箱出口之前，海关需对集装箱中的货物进行随机抽查。这种原因导致的翻箱操作难以避免，属于码头可控制的范围之外。移箱翻箱一般为小概率事件，虽然可能导致无效作业时间增加，但对港口服务效率一般不会产生太大影响。

装船翻箱是指由于送箱时出口箱在堆场中安排不合理或配载计划安排不合理，在集装箱装船时为了尽量保持和配载图一致而发生的翻箱作业。优化装船前预翻箱方案能最大程度减少装船时翻箱的可能，有利于提高装船速度，但并非所有的集装箱码头都有足够的能力和时间为每条船舶的出口箱进行装船前预翻箱整理。

提箱翻箱是指外集卡进入堆场提箱时，由于客户提取的目标箱堆存在下层，

上层集装箱尚未提走，产生压箱，此时需要将上方的阻塞箱移动到其它栈。这种翻箱对码头进口箱的提箱作业效率有一定影响，造成作业成本上升。

2.3.2 进口箱翻箱原因分析

根据相关法律规定，动物检验检疫、海关等各相关部门对进口集装箱进行抽检时的翻箱难以避免且为小概率事件，本文不考虑此类翻箱。进口箱在提箱时产生翻箱的根本原因在于货主的提箱顺序与堆场内相关集装箱的作业顺序不一致。提箱顺序取决于货主的预约提箱时间，而集装箱在卸船时的箱位分配方案决定了集装箱在堆场的作业顺序，这两个因素共同决定了翻箱量。

(1) 卸船时形成的翻箱问题

船舶到港前堆场在为集装箱分配箱位时，进口箱一般遵循“FIFO-先进先出”的原则，最理想的情况是新卸下的集装箱不压在堆场中未提走的集装箱上，但在实际作业中，由于堆场使用率较高，实际卸船作业时堆场中可能没有合适的空位或空栈，一些司机为图方便将新卸下的箱子压在堆场原有的集装箱上，这种放箱行为势必会造成提箱作业时的翻箱。

(2) 货主提箱时形成的翻箱问题

货主提箱时翻箱是由于外集卡进入堆场提箱时，客户的提箱顺序与卸船时集装箱在堆场中的作业顺序不匹配导致，客户提箱时间比较随机，码头方很难在卸船前完全掌握这批箱子的提箱时间。此外，如果在有翻箱的情况下，对阻塞箱落箱位置安排的不合理，也会引起后续提箱过程的再次翻箱。

2.3.3 降低进口箱提箱时翻箱的方法

在实际堆场翻箱的控制过程中涉及到很多方面的共同协作，本文提到的翻箱的预防措施主要针对减少货主提箱时的翻箱，然而对此类翻箱的控制首先要从集装箱卸船时开始。

(1) 大客户分类法。在实施这种方法前首先要对以往企业进口箱的数量规律进行梳理，将进口箱数量较多的货主定义为堆场的VIP客户。卸船前根据卸船清单寻找到每一个VIP客户的进口箱，根据VIP进口箱的数量合理分配放箱位。由于是同一货主的集装箱，提箱时不分先后，好的规划方案能够做到提箱时实现零翻箱，这种方法在降低堆场成本的同时，提高了客户的满意度。

(2) 采取预约提箱机制。这一机制要求船公司在到港之前尽可能提供集装箱的预离港时间，即进口箱的提取时间。由于靠港船公司数量较多，当前港方并未强制船公司必须提供提箱时间。即使如此，当货主没能在预约时间内提走进口箱时，若集装箱仍在免费堆存期内，港方也未有惩罚措施，这给堆场分配进口箱箱位时带来了不利影响，严重干扰了箱位分配作业计划的实施，降低了后续提箱作业的效率，导致了許多不必要的翻箱发生。因此码头应提出一个奖惩机制，使货主尽可能在预约时间内提箱。

(3) 优选提箱法。进口重箱普遍存在“一票多箱”的情况，同一提单下的多个集装箱一般堆存在后方堆场的不同位置。采用优选提箱法就是对这类箱的提箱顺序不分先后，先提堆存在最上层的集装箱，这样能够做到减少翻箱。

要减少翻箱应该从全局出发，协调码头及船舶各方面，保持信息的畅通，共享信息化的成果，在制定计划时，要权衡各方面的利益，尽量使各个部门满意。

2.4 本章小结

本章首先介绍了集装箱码头的主要布局以及堆场的主要业务内容，接下来将各大堆场目前采用的堆存规则进行了分析，同时还介绍了一下堆场中箱位编号的基本知识，在本章最后，针对本文研究的问题，重点分析了集装箱在卸船过程以及货主提箱过程中产生翻箱问题的原因，并且结合集装箱码头在长期堆场管理中积累的经验，针对两个过程提出了降低翻箱问题的方法。

第3章 集装箱堆场卸船时箱位分配问题优化模型

3.1 进口箱堆存优化问题描述

3.1.1 问题描述

进口箱在卸船时箱位分配过程中,对于卸船或者直接进港到达后方堆场的集装箱,堆场计划人员会凭据其个人经验将集装箱指定堆放到堆场中某个箱位^[31],在放箱时一般按照不同提单、不同货主的集装箱分开堆存在不同贝位的规则。这种安排很明显能够减少货主提箱时的翻箱量,然而在此规则下,堆场需要预留很大的堆存面积。随着近些年集装箱港口吞吐量的不断增加,如果依照原来的箱位分配策略,现有堆场面积很难满足要求。出于多方面的考虑,现有堆场通常采用不同货主的进口箱混合堆存的策略,然而无论怎么堆存,在这种情况下都很容易产生翻箱。各大港口近些年都加快了自身的信息化建设,信息共享机制方便了港口和货主,预约提箱机制能够让港口提前获悉进口箱提取的时间,成为码头进行箱位分配时的主要参考因素。在实际作业中,集装箱进出场的顺序、在堆场中的堆存状况以及提箱翻箱时阻塞箱落箱位置的选择共同影响着集装箱堆场的翻箱量^[32]。假设在卸船作业开始前,码头方已经获得了进口箱的部分信息,其中包括卸船顺序和货主的预约提箱时间段,并且堆场中预留给此批进口箱的放箱位置是空贝,考虑到作业的便利性和较小的机械损耗,堆场规定提箱过程中需要翻箱时只能在同贝内选择阻塞箱的落箱位置。图3.1a所示为部分集装箱在船上的堆存位置,图3.1b为堆场计划人员给这些进口箱在堆场中分配的相应的堆存位置。由于堆场机械作业能力有限,堆场中一般采用6栈4层的堆存方式,混合堆存的作业规则使得在多数情况下提取集装箱时都需要翻箱。为了方便翻箱,每一贝都留2~3个空箱位。本文所做的研究主要是在卸船顺序和提箱时间确定的情况下通过安排具体的堆存箱位以及翻箱时的落箱位置,使堆场能够通过最少次数的翻箱来完成货主的提箱过程。

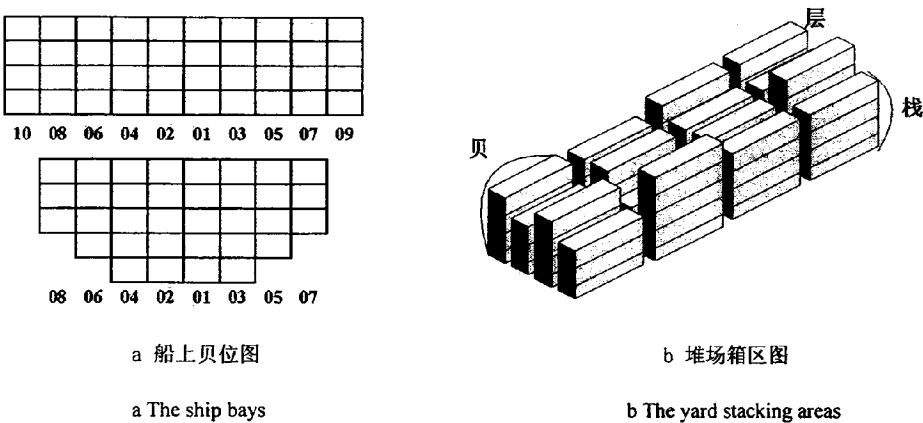


图 3.1 船上贝位及对应堆场堆存状态图

Fig. 3.1 Bays on the ship and the corresponding yard stacking areas

3.1.2 问题分析

(1) 预约提箱机制分析

本文采取不同提单不同货主的进口箱混合堆存的策略，以集装箱的提箱时间为约束来制定进口箱堆存计划。在确定提箱时间时，本文采用了一种预约提箱机制。基于预约提还箱系统在许多码头已经得到了推广，文中假定所有货主在集装箱到港前已经对提箱时间做好预约。实际作业中，港口规定了进口箱的免费堆存期限，货主一般在集装箱到港的免费堆存时间内提箱，不同港口其免费堆存期限不同。本文对一批进口箱进行箱位分配，由于本批进口箱数量较少，假定在一天内本批箱全部被提走，并且提箱的开始时间为 0 点，选择 24h 作为箱位分配的决策周期，为了方便确定集装箱提箱时的先后顺序，本文在对预约提箱时间进行处理时，将提箱时间转化为提箱优先级。表 3.1 很清晰表明了提箱时间与提箱优先级之间的映射关系, 映射值越小表示集装箱被提走的时间越早, 在堆存时应尽量安排在上层。另外在影响集装箱提箱翻箱量的因素中，提箱目的地距离堆场的远近也在一定程度上影响着集装箱的提取顺序。为方便本问题的研究，本文忽略目的地的影响，仅用集装箱的提箱优先级来表示提箱时刻的早晚，并且认为具有同一提箱优先级的集装箱为同一货主箱，在提箱时不分先后。

表 3.1 集装箱提箱时间段与提箱优先级映射表

Tab. 3.1 The relationship between time pick-up containers and the priority

提箱时段	0: 00-1: 00	1: 00-2: 00	2: 00-3: 00	...	23: 00-24: 00
提箱优先级	1	2	3	...	24

(2) 堆存方案分析

在集装箱进场时,不同的堆存方案会产生不同的翻箱量,下面简单取 18 个进口箱进行箱位分配来说明堆存计划对翻箱量的影响。堆场相应的在 1 个空贝位中为这些集装箱安排堆存位置,同时假设集装箱为 1~18 号依次入场,这几个进口箱对应的提箱优先级如表 3.2 所示。

表 3.2 待入场集装箱提箱优先级

Tab. 3.2 Import containers pick-up priority

箱号	1	2	3	4	5	6	7	8	9
优先级	15	22	4	23	19	23	13	8	19
箱号	10	11	12	13	14	15	16	17	18
优先级	19	16	21	3	16	24	4	6	7

同时堆场给这 18 个进口箱随机安排了两种堆存方案,见图 3.2。

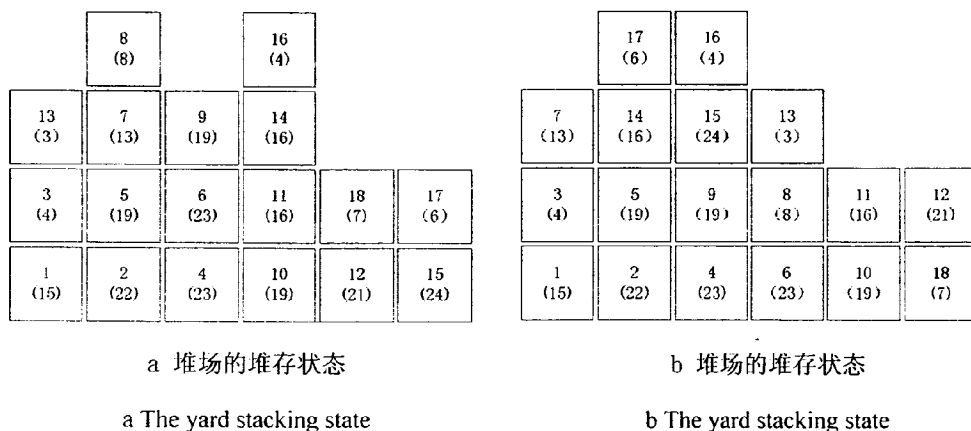


图 3.2 进口箱在堆场的堆存状态

Fig. 3.2 The yard stacking state of the import containers

在上图所示的堆存状态下,每个括号中的数字代表对应集装箱的提取优先级。13 号箱优先级最高,为当前状态下的目标箱。在图 3.2a 中,货主依照提箱优先级由小到大的顺序取箱,直至取完 15 号箱,整个提箱过程都不会产生翻箱。

在图 3.2b 这种堆存状态下, 首先提取目标箱 13, 下一个状态提取目标箱 3 或 16。当提取 3 号箱时 7 号箱为阻塞箱, 需要对 7 号箱进行翻箱操作, 翻箱时 7 号箱有 3 个放箱位置①②③, 如图 3.3, 在这里我们给 7 号集装箱安排的是对整个贝翻箱影响最小的位置, 例如选择位置②, 这样就防止了二次翻箱。

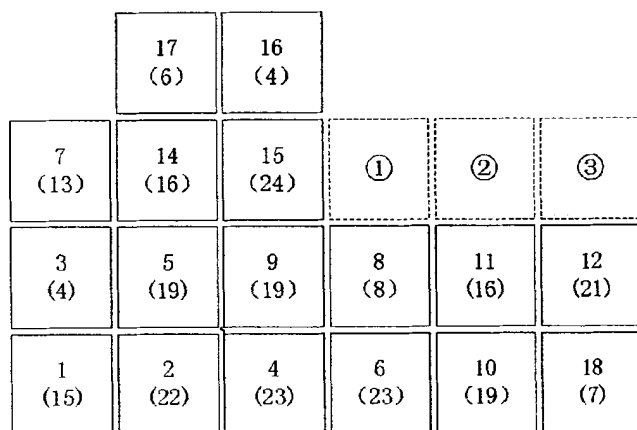


图 3.3 某时刻集装箱的堆存状态

Fig. 3.3 The stacking state at a time in container yard

(3) 翻箱时落箱位置分析

为了减少二次翻箱, 根据相关研究在确定阻塞箱的落箱位置时按照以下步骤进行:

① 若当前堆存状态下该贝内有空栈, 阻塞箱的落箱位置首选离其最近的空栈。

② 若该堆存状态下不存在空栈, 在同一贝中寻找离其最近的满足栈中每个集装箱的提箱优先级都大于阻塞箱的有放箱位置的栈。

③ 若没有出现①、②两步中的栈, 则在该贝位内选取最矮栈进行放箱, 优选距离阻塞箱最近的。

下面以图 3.2b 为例模拟翻箱时阻塞箱落箱位置的选取过程。在移动阻塞箱时, 优选的落箱位置是落于某一栈上不会增加后续提箱时翻箱量的箱位, 如果没有这样的目标栈, 则选择一个最矮栈。在图 3.2b 中, 目标箱为 13 号箱且位于最上层, 则 13 号箱首先被提走。在下一个状态中, 目标箱为 3 和 16, 由于 16 号箱在最上层, 优先提取, 更新堆存状态图如 3.4 所示。

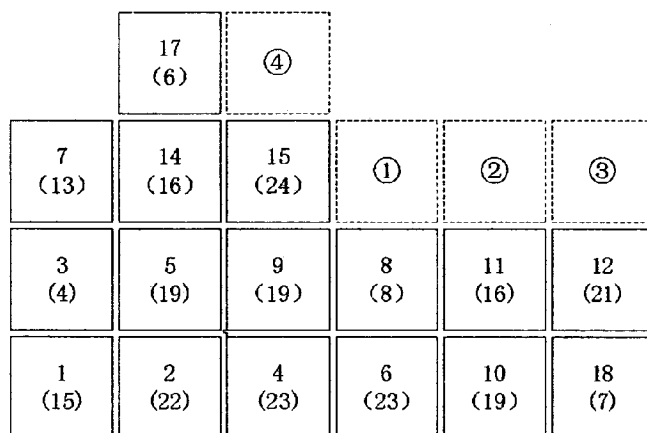


图 3.4 堆场中某贝更新后的堆存状态

Fig. 3.4 The stacking state at the next time

在图 3.4 中，3 号箱为该状态下的目标箱，其上有 1 个阻塞箱，首先对 7 号箱进行翻箱操作。根据上述翻箱步骤，7 号箱放于②或④后都可以保证其不会成为新的阻塞箱，选取②为落箱位后更新堆存状态见图 3.5。

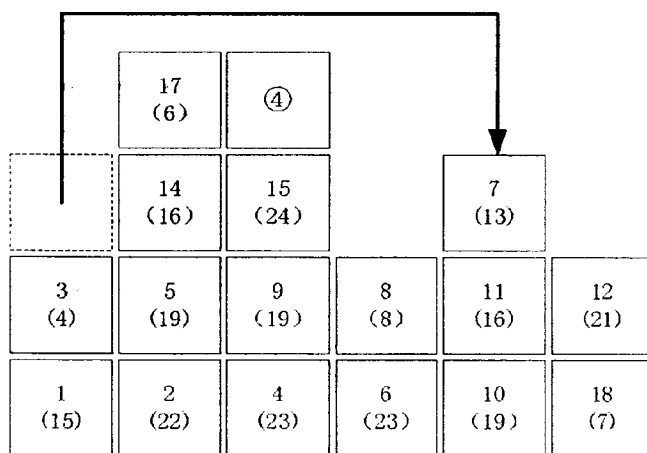


图 3.5 某贝下一堆存状态图

Fig. 3.5 The stacking state in a bay at a time

由上面仿真过程可知文中翻箱时确定放箱位置的方法在一定程度上能避免二次翻箱。

通过对提箱翻箱问题的分析可知：进口箱在卸船顺序一定的条件下，不同的箱位分配方案会产生不同的翻箱操作，进而影响提箱时的翻箱量，所以为了减少

翻箱，可以从优化进口箱在堆场初始的箱位分配开始。

3.2 进口箱堆存优化模型

3.2.1 模型假设

影响卸船时间的因素很多，如果全部考虑，模型必将会变的很复杂。为了研究卸船堆存问题的实质，本文选择一个箱区的作业进行研究并作了如下假设：

- (1) 本文所考虑的集装箱都是 20 英尺长的标准通用集装箱，且如果有翻箱的情况发生，翻箱在同一贝位进行作业；
- (2) 卸船顺序已知，堆场预留给这批箱的放箱贝位已定且初始状态为空；
- (3) 到港前船上所有进口箱的提箱时间都被预约，且在规定的时段被提走；
- (4) 卸船过程中所涉及到的贝位不发生提箱操作；
- (5) 此批箱在未全部提走前，不允许新的箱子进入。

3.2.2 建立数学模型

(1) 主要符号以及变量说明

m, n : 需要卸载的集装箱编号; $m, n = 1, 2, \dots, N$ $m \neq n$;

i : 进口箱预分配到堆场中所在贝的编号; $i = 1, 2, \dots, I$;

j : 进口箱预分配到堆场中所在栈的编号; $j = 1, 2, \dots, J$;

k : 进口箱预分配到堆场中所在层的编号; $k = 1, 2, \dots, K$;

c_n : 进口箱 n 的取箱优先级; $c_n = 1, 2, \dots, c$;

L_n : 集装箱 n 的卸船（入场）顺序; $L_n = 1, 2, \dots, N$;

S_n : 集装箱 n 的作业顺序; $S_n = 1, 2, \dots, S$;

H_{ij} : i 贝 j 栈的栈高; $H_{ij} = 1, 2, \dots, K$;

c_{\min} : 对于这批进口箱，某集装箱的提箱优先级是在此状态下最小的;

R_{nij} : 提取位于 (i, j, k) 位置的目标箱 n 时的翻箱量;

V_i : 表示堆场中第*i*贝的容量;

V : 卸船时安排到某一箱区的集装箱量;

M : 无穷大的正数;

决策变量

$$X_{nijk} = \begin{cases} 1 & \text{需要卸载的集装箱} n \text{分配到堆场中的位置为}(i, j, k) \\ 0 & \text{其它} \end{cases}$$

(2) 模型建立及说明

目标函数

$$\text{Min } Z = \sum_{n=1}^N R_{nijk} \quad (3.1)$$

约束条件

$$R_{nijk} = H_{ij} - k \quad \text{if } c_n = c_{\min} \quad (3.2)$$

$$\sum_{n=1}^N \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K X_{nijk} = V \quad (3.3)$$

$$I - 1 \leq \frac{N}{V_i} \leq I \quad (3.4)$$

$$\sum_{n=1}^N \sum_{j=1}^J \sum_{k=1}^K X_{nijk} \leq V_i \quad \forall i \quad (3.5)$$

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K X_{nijk} = 1 \quad \forall n \quad (3.6)$$

$$\sum_{n=1}^N X_{nijk} \leq 1 \quad \forall i, j, k \quad (3.7)$$

$$\sum_{m=1}^N X_{mijk} - \sum_{n=1}^N X_{nijk'} \leq 0 \quad \forall i, j, k > k' \quad (3.8)$$

$$M \cdot L_m \cdot X_{mijk} > L_n \cdot X_{nijk'} \quad k < k' \quad (3.9)$$

$$\begin{aligned} &\text{if } X_{mijk} = 1, X_{nijk'} = 1, k < k' \\ &\text{then } S_m < S_n \end{aligned} \quad (3.10)$$

在模型中, 目标函数表示不同货主根据预约时间提取完所有集装箱时相应堆

场内发生的总翻箱量最小。约束条件(3.2)表示提取某一集装箱时,堆场发生的翻箱量,且被提取集装箱在当前状态下其提箱优先级是最小的;约束条件(3.3)表示待卸载集装箱被分配到某一箱区的进口箱总数与堆场为进口箱在此箱区分配的箱位数相等;约束条件(3.4)表示根据进口箱的规模,在堆场中分配的贝位数;约束条件(3.5)为贝位内集装箱容量限制;约束条件(3.6)限定一个箱子只能分配到一个位置上;相应的约束条件(3.7)说明一个箱位最多只能堆放一个集装箱;约束条件(3.8)表示堆场的集装箱不能悬空放置,即对于同一栈的集装箱,必须先放置下面的集装箱,然后才能放置上面的集装箱;约束条件(3.9)表示同一栈中下层的集装箱要比上层的集装箱早入场;约束条件(3.10)说明在对集装箱进行作业时,只能对最上层的集装箱进行操作。

集装箱堆场箱位分配问题属于组合优化问题,对于这类问题,约束条件种类繁多,随着问题规模的增加,组合优化问题的搜索空间也急剧扩大,利用经验得到的分配方案会带来很大的翻箱,有时在计算机上也很难求解最优值。遗传算法经过多年的研究和实践,在解决复杂系统优化问题时已显示出其良好的求解运算能力,特别是对一些 NP 组合优化问题的求解,更表现出了其优越的性能。遗传算法在求解旅行商问题、背包问题、装箱问题等方面均有很多成功运用,实践证明,遗传算法在求解组合优化中的 NP 问题时,运算效果好、适用广泛,是一种非常有效的工具。遗传算法在解决 NP 问题时所表现出的杰出的运算能力给本文模型的求解提供了参考,本文将尝试采用遗传算法对进口箱箱位分配问题求解。

3.3 本章小结

提箱时进口箱的翻箱问题一直困扰着码头管理者,本文基于这一实际问题,针对前文对卸船提箱过程所做的分析,在集装箱的卸船顺序和提箱时间确定的情况下以提箱时的翻箱量最小为目标,为进口箱分配入场箱位。当需要翻箱时,为了避免二次翻箱,提出了一套完整的翻箱策略。由于集装箱箱位分配问题属于组合优化问题,考虑到遗传算法在解决这类问题时表现出的杰出的运算能力,在后面的章节决定用遗传算法求解此类问题。

第4章 进口箱箱位分配问题算法设计

4.1 遗传算法概述

4.1.1 基本思想

遗传算法(Genetic Algorithm, 简称 GA)是由美国 Michigan 大学的 Holland 教授(1969)在借鉴生物界自然变异和自然遗传机制的基础上提出的一种随机化搜索算法, 后经由 De Jong(1975), Goldberg(1989)等归纳总结所形成的一类模拟进化算法。GA 摒弃了传统的搜索方式, 通过模拟自然界生物进化过程, 采用人工进化的方式对目标空间进行随机优化搜索。它将问题域中的可行解看作是群体中的一个个体或染色体, 并将每一个个体编码成符号串形式, 模拟达尔文的遗传操作和自然淘汰的生物进化过程, 对群体反复进行基于遗传学的操作(遗传、交叉和变异)。根据预定的适应度函数对每个个体进行评价, 依据适者生存、优胜劣汰的进化规则, 不断得到更优的群体, 同时以全局并行搜索方式来搜索优化群体中的最优个体, 以求得满足要求的最优解。作为一种实用、高效、鲁棒性强的优化技术, 其发展极为迅速, 现已广泛应用于函数优化、组合优化、自动控制、图像处理等科技领域, 成为现代有关智能计算中的关键技术。

4.1.2 基本步骤

遗传算法寻优的方法是模仿自然界生物的遗传进化过程, 即把问题的解通过简单的编码技术表示成染色体, 并基于适应度值来选择染色体。遗传算法主要使用选择算子、交叉算子与变异算子来模拟生物进化, 从而使种群延续下去。遗传算法的基本步骤如下:

(1) 编码

遗传算法的编码决定了问题的可行解在染色体中的表现形式, 在应用遗传算法时, 首先要找到一种合适的编码方式将解映射到染色体中, 这也是设计遗传算法时的一个关键步骤。在遗传算法执行过程中, 对不同的问题进行编码, 编码的好坏直接影响选择、交叉、变异等遗传运算。从理论上讲, 编码的目的是为了更好的解决问题而不是简单的说明问题。

(2) 初始种群的生成

生成初始种群需要考虑两个问题：一是生成方法的选取，二是种群规模的确定。为了保证种群的进化尽可能遍历所有的状态，寻找到全局最优解，初始种群一般随机选取。然而初始种群的随机选取增大了种群的进化代数，基于一定规则的择优选取法能够降低这种随机性。在确定种群规模时，过大会增加适应度值的计算量，过小会导致迭代次数增多，而且容易陷入局部极小点，要根据实际问题合理选取种群规模。

(3) 适应度评估

适应度表明了个体或解的优劣性，适应度值一般来讲与个体遗传到下一代的概率成正比。度量个体适应度的函数称为适应度函数，针对不同的问题，适应度函数的定义方式也不同。适应度函数总是非负的，任何情况下希望其取值越大越好，然而目标函数有正有负，经常需要在这两个函数之间进行转换^[33]。

(4) 选择

选择是在群体中选择生命力强的个体产生新的群体的过程，这一过程体现了达尔文的适者生存原则。选择的目的是从群体中选出较适应环境的个体，并让这些选中的个体再去繁殖下一代。选择一般采用两种策略，第一种称为精英保留策略，第二种是轮盘选择策略。当想将两个较优的个体的好的基因完全传给下一代时，采用精英保留策略；当想在群体中随机选出两个个出众的个体并通过一定的策略进行处理繁殖下一代时，选用轮盘选择策略。

(5) 交叉

遗传算法中所谓的交叉运算，是指对两个相互配对的染色体按照某种方式相互交换某部分基因，从而形成两个新的个体^[34]。交叉运算是遗传算法区别于其他进化运算的重要特征，在遗传算法中起着关键作用。新个体产生的方法主要是在交叉运算中完成，它决定了遗传算法的全局搜索能力。

(6) 变异

变异首先在群体中随机选择一个个体，对于选中的个体以一定的概率随机的改变染色体中某个基因的值。同生物界一样，遗传算法中发生变异的概率极低，通常取值很小。在遗传算法中使用变异算子主要有以下两个目的：改善遗传算法的局部搜索能力；维持群体的多样性，防止出现早熟现象。交叉算子提高了全局

搜索能力,变异算子提高了局部搜索能力,这两种算子相互配合使得遗传算法无论是全局搜索还是局部搜索都很出色。

遗传算法的主要过程如下:

Begin

- ① 初始化参数: 如种群规模 M , 进化世代数 T 等。
- ② 初始种群: 采用随机的方式产生 M 个个体作为初始群体 $P(0)$
- ③ 计算适应度: 计算初始种群中各个个体的适应度值, 对各个个体进行评价。
- ④ 选择运算: 先计算各个体的选择概率, 根据轮盘赌方法在当前种群中进行选择操作。
- ⑤ 交叉运算: 对选择好的双亲运用交叉算子产生子个体。
- ⑥ 变异运算: 对产生的子个体以很小的概率运用变异操作。

群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$ 。

- ⑦ 判断终止条件, 若 $t \leq T$, 则: $t = t + 1$, 转到③; 若 $t > T$, 则转到⑧。

- ⑧ 输出遗传算法进化过程中所得到的具有最大适应度的个体。

End

4.2 进口箱堆存优化模型算法设计

4.2.1 染色体的编码及初始解的产生

编码的目的是将问题的可行解转换到遗传算法所能处理的搜索空间中去, 对于实际问题, 编码要考虑到后续的交叉变异以及对于最优解的解码。二进制编码方式在遗传算法中被经常使用, 每个变量都表示成二进制的形式, 编码解码操作简单易行, 后续的交叉变异等遗传过程也便于实现。然而本文所要解决的堆场箱位分配问题由于条件特殊, 使用二进制编码或者浮点数编码会使问题更加复杂, 而采用实数编码则不但能更加形象表达问题, 且大大缩短编码长度, 同时不用解码, 方便计算。

为了便于表示进口箱的相关信息, 首先设计一个矩阵 $B(n, l, c, x)$, n 代表集装箱的箱号, l 代表集装箱的卸船顺序, c 代表集装箱的提箱优先级, x 代表堆场为

进口箱分配的放箱位置。 B 中每一行都是染色体的一个基因，所有行构成一条染色体。由前文可知，要确定集装箱在某一箱区中的具体位置需要三个参数：贝、栈、层。为了方便描述集装箱在堆场中的位置，图 4.1 给出了某一贝内箱位的截面图。

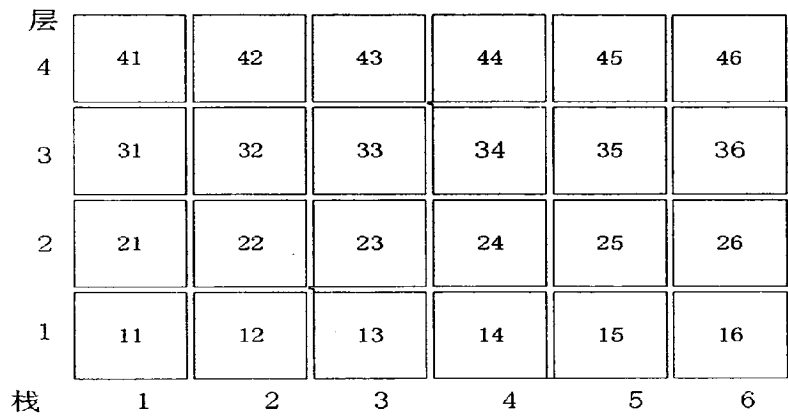


图 4.1 堆场中某一贝截面图

Fig. 4.1 The section diagram of a bay in the container yard

每个箱位的编号如图所示，编号的个位代表集装箱所在的栈，十位代表集装箱所在的层。 $x(11, 21, 31, 41, 12, 13, 14, 23)$ ，表示第一个集装箱进入 11 号箱位，第二个集装箱进入 21 号箱位，第三个集装箱进入 31 号箱位，以此类推。本文在对集装箱的放箱位置进行编码时为了避免出现悬空箱以及简化箱位表示，采用如下转换： $x(11, 21, 31, 41, 12, 13, 14, 23) \rightarrow x(1, 1, 1, 1, 2, 3, 4, 3)$ 。变换后的编码

$x(1, 1, 1, 1, 2, 3, 4, 3)$ 表示第一个集装箱放到第 1 栈，第二个集装箱也放到第 1 栈，由于第 1 栈已经放箱，因此第二个集装箱对应的箱位号为 21 号，第三个集装箱对应的箱位号为 31 号。通过分析可知转换后的编码方式同样能够表示集装箱在堆场中的位置，此编码方式对多贝位的情况同样适用。当对多贝位放箱位置进行编码时，将 n 个贝位并列排放，形成 $6n \times 4$ 的放箱矩阵。例如当堆场中提供 2 个贝位作为放箱位时，第 1 个贝位的栈记为 $x(1, 2, 3, 4, 5, 6)$ ，第 2 个贝位的栈号依次变为 $x(7, 8, 9, 10, 11, 12)$ 。 $x(7, 10, 11, 8, 3)$ 则表示进口箱分别放到第 2 贝的第

1, 4, 5, 2 栈以及第 1 贝的第 3 栈。结合上文对堆存位置编码方式的介绍, 下面对矩阵 B 进行说明。

$$\text{在矩阵 } B = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 19 & 6 & 9 & 17 & 24 & 18 & 4 & 18 \\ 1 & 1 & 10 & 3 & 9 & 8 & 3 & 9 \end{pmatrix} \text{ 中, 第1行表示集装箱的序号, 第2行}$$

表示集装箱的卸船顺序, 第3行表示相应集装箱的提箱优先级, 第4行表示为这些集装箱分配的放箱位置。由于矩阵 B 前3行信息已知, 因此在后文进行遗传操作时, 只需对 x 基因即堆场为进口箱分配的位置进行操作。

初始种群的生成是遗传算法中重要的一步, 为了得到较好的初始群体, 加快算法的求解速度, 本文采用如下方法产生初始种群: 首先生成 $2M$ 个初始可行解, 然后通过计算目标函数选择较好的 M 个可行解作为问题的初始种群。

4.2.2 适应度函数的设计

适应度函数作为解得评价指标, 是根据目标函数确定的用于区分群体中个体好坏的标准, 其选择对于利用遗传算法求解问题显得至关重要, 适应度函数将会直接影响到遗传算法的收敛速度以及能否寻求到最优解。在设计适应度函数的时候应尽量做到简单有效。在遗传算法中适应度函数的值必须是非负的, 目标函数并不一定能作为适应度函数, 在遗传算法执行时, 有时需要灵活的对目标函数进行变换, 即适应度的尺度变换。

文中采用动态方法, 考虑到提箱翻箱时落箱位置对后续提箱的影响, 每一个提箱阶段只记录当前目标箱的翻箱量。本文设计的目标箱翻箱量的计算方法如下: 记录目标箱所在层为 j , 所在栈的高度为 h , 则 $h-j$ 代表提取此目标箱时发生的翻箱量。当 $h-j > 0$ 时表示目标箱上方有阻塞箱, 根据下文的翻箱规则进行翻箱操作, 并更新堆存状态。

考虑到本文数学模型的目标函数是求翻箱量的最小值且结果都为非负, 在此不妨将适应度函数表示成目标函数倒数的形式, 同时为了避免分母出现0的可能性,

用 $fitness = \frac{1}{1+Z}$ 这种转换方法。这样就保证了目标函数的优化方向对应于适

应度值增大的方向,选择的时候,翻箱量较小的堆存方案遗传到下一代的概率变大。

4.2.3 遗传算子的确定

(1) 选择算子的确定

从群体中选择优胜个体,淘汰劣质个体的操作被称为选择。选择的目的是优胜劣汰,即适应度较高的个体有较大的概率直接被遗传到下一代群体或者通过交叉产生新个体再遗传到下一代。进口箱堆场箱位分配问题模型是在实际知识背景的基础上建立起来的,求解出的个体适应度值之间的差异一般情况下比较小,因此,考虑到目前常用几种选择算子的优缺点,本文采用最基本最常用的选择方法即轮盘赌进行选择操作。在该方法中,各个个体被选中的概率与其适应度成正比。设群体规模大小为 M ,个体 i 的适应度函数值为 f_i ,则这个个体被选择的概率为:

$$P_{si} = f_i / \sum_{i=1}^M f_i。显然,个体适应度越大,其被选择的概率越高。$$

(2) 交叉算子的确定

编码方式决定了交叉算子的选择,对于实数编码的遗传算法在交叉算子的设计上目前没有统一的标准。单点交叉又称简单交叉,它将选择后的个体进行两两配对,然后随机选择一个交叉点,对参与交叉操作的两个个体,交换交叉点以后的部分。由于其操作简单,本文主要采用单点交叉的方法来进行染色体之间的交叉。下面的例子是一个单点交叉:

父代A: 11112 ↑ 343 → 11112 241 (子代A)

父代B: 12143 ↑ 241 → 12143 343 (子代B)

经过单点交叉,染色体对应的堆存方案有可能会超箱(如子代A中有5个集装箱堆存在第1栈),可以通过适当修改将其变成可行解。对于超高的染色体基因,修改的方法是观察所有基因出现的次数,找出出现次数过多的基因,随机用出现次数较少的基因取代;每取代一次,更新各个基因出现的次数值;直到没有基因出现的次数过多,染色体成为可行解。

(3) 变异算子的确定

遗传算法中的变异操作是产生新个体必不可少的方法,关于变异的操作方法

是：在 x 基因中随机选取一个变异点，根据条件判断是否交换从此变异点开始直到 x 基因末端这之间的相邻基因的位置。由于本文采用了实数编码方法，对于 M 个待卸箱（编号从 1 到 M ）来说，变异点只能从 M 个数中选取。在进行每一代遗传操作时，随机从 1~ M 之间选取变异点 N ，从 N 开始，依次判断第 N 和 $N+1$ 个箱是否在不同栈，如果在不同的栈，那么这两个箱子符合交换的条件，是否交换要看第 N 和 $N+1$ 个位置的基因交换后此染色体对应的总的翻箱量是否减少，若减少，则进行交换，否则保持原基因位置不变，直到 $N=M-1$ 。

4.3 本章小结

本章首先对遗传算法进行了简单的介绍，其中包括遗传算法的基本思想和基本步骤，然后针对本文的问题，选择合适的编码及遗传算子进行模型的算法设计，在进行交叉以及变异时本文设计了特殊的交叉变异算子。

第 5 章 算例分析

5.1 算例背景

提箱作业在任意时间段都有可能发生，并且经常出现不同箱区同时作业的情况，然而集装箱堆场各箱区的提箱作业是相互独立的，因此以堆场中的某一箱区为例进行模拟和计算，也具有较好的代表性。

假设船上某贝位进口箱的卸船顺序如表5.1所示，并且堆场在某一箱区内给这些集装箱分配了2个空贝位用于安排放箱，要求根据这40个集装箱的卸船顺序以及提箱优先级确定一个合理的后方堆场堆存方案，使得提箱时总的翻箱量最小，在此合理假设卸船顺序即集装箱序号。

表 5.1 集装箱的卸船顺序以及提箱优先级

Tab. 5.1 The unloading sequence of container and the class of extracting time

卸船顺序	1	2	3	4	5	6	7	8	9	10
提箱优先级	19	6	9	17	24	18	4	18	7	5
卸船顺序	11	12	13	14	15	16	17	18	19	20
提箱优先级	22	8	5	19	10	22	14	24	2	2
卸船顺序	21	22	23	24	25	26	27	28	29	30
提箱优先级	15	9	7	14	6	12	6	10	15	3
卸船顺序	31	32	33	34	35	36	37	38	39	40
提箱优先级	10	17	1	10	1	4	12	13	16	12

5.2 算例求解

在集装箱卸船堆存过程中，每个进口箱都有多个可候选的箱位及落箱位置，由前文可知不同的堆存状态以及翻箱操作时落箱位置的不同选择共同影响着货主取箱时的翻箱量，本文用一个具体算例来验证文中遗传算法的有效性。

当进口箱数量为40时，本文通过遗传算法对此算例进行优化，在优化时设定初始种群规模为80，遗传代数数为100代，在10次遗传操作中有8次结果为3，利用本文的遗传算法为其分配的堆存方案如表5.2所示：

表 5.2 40 个集装箱的堆存方案

Tab. 5.2 The storage scheme of 40 containers

入场序号	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25 -26-27-28-29-30-31-32-33-34-35-36-37-38-39-40
提箱优先级	19-6-9-17-24-18-4-18-7-5-22-8-5-19-10-22-14-24-2-2-15-9-7-14-6-12 -6-10-15-3-10-17-1-10-1-4-12-13-16-12
堆存位置	1-1-10-3-9-8-3-9-5-7-11-2-7-12-11-6-5-4-1-1-6-10-2-4-2-5-10-9-8-3 -10-12-11-6-5-6-4-8-12-12
提箱顺序	35-33-20-19-30-7-36-13-10-25-2-27-23-9-12-22-3-15-28-34-31-26-37- 40-38-17-24-29-21-39-32-4-6-8-14-1-16-11-18-5

上述方案对应的收敛图如图 5.1 所示：

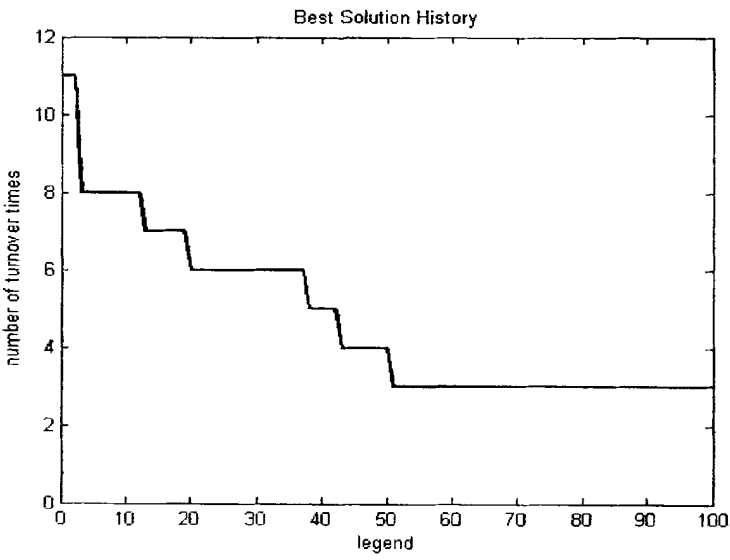


图 5.1 最小翻箱量收敛图

Fig. 5.1 The convergence diagram of minimum relocation

从图 5.1 可以看出提取集装箱时的翻箱量随着遗传代数次数的增加而逐渐较少，直到收敛到最优解，对应地的翻箱量为 3，一直到第 100 代，目标函数最优解的值也没有变化，表明此时求得的结果较优。此时堆存方案对应应在堆场中的堆存状态见图 5.2：

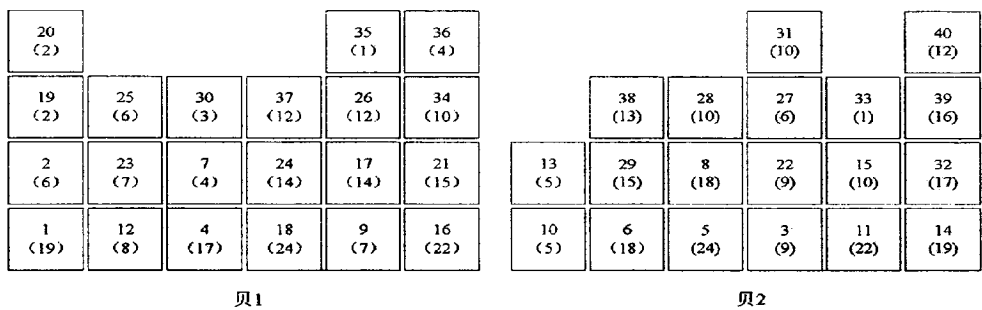


图 5.2 40 个集装箱最优堆存状态

Fig. 5.2 The best storage location of 40 containers

由收敛图可知，在上述堆存状态下提箱时需要翻箱，在此给出翻箱方案。货主依照预约提箱时间进行提箱作业，当堆场堆存状态如图 5.3 时，

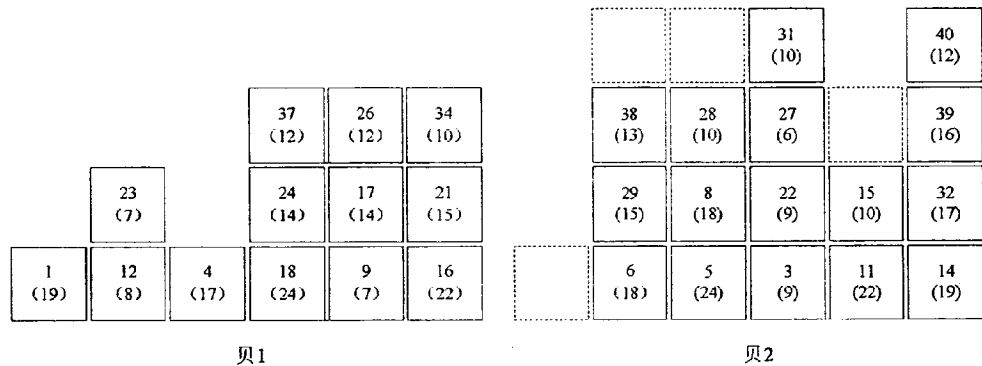


图 5.3 提箱过程中堆存状态

Fig. 5.3 The state of storage location in the process of extracting containers

在图 5.3 这一状态下的目标箱为 27 号箱，可以看到目标箱 27 上面有阻塞箱 31，则需要把阻塞箱移到同贝内其它栈，翻箱示意图如图 5.4 所示。

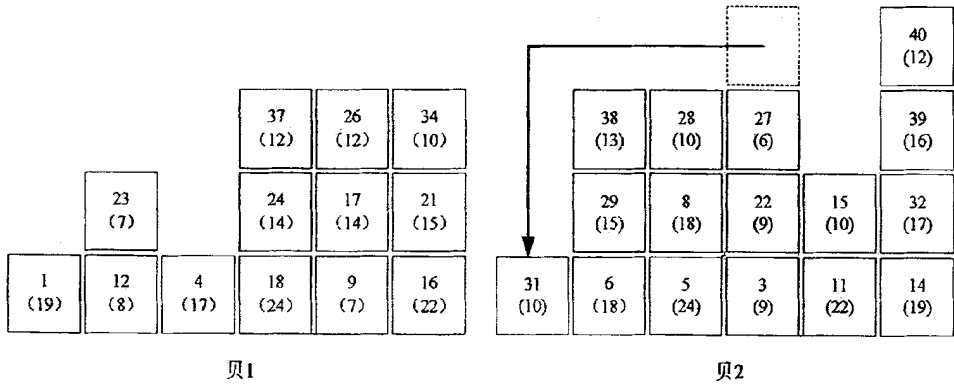


图 5.4 提取 27 号箱时的翻箱方案

Fig. 5.4 The relocation plan in the process of extracting the 27th container

提箱作业依据提箱顺序继续进行，当堆场堆存状态如下时，见图 5.5。

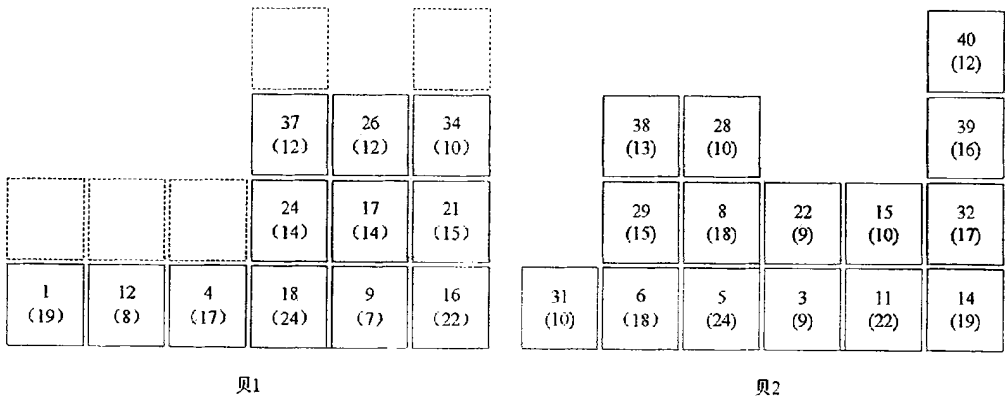


图 5.5 提箱过程中堆存状态

Fig. 5.5 The state of storage location in the process of extracting containers

此刻需要提取的目标箱为 9 号集装箱，则需要把阻塞箱 17、26 移到同贝内其它栈，翻箱后的状态如图 5.6 所示。

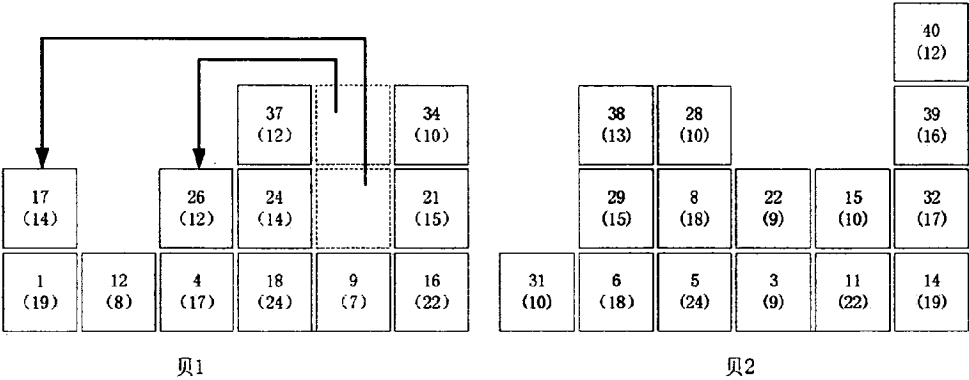


图 5.6 提取 9 号箱时的翻箱方案

Fig. 5.6 The relocation plan in the process of extracting the container 9

5.3 不同规模下的算例求解分析

考虑到前文优先级设定时仅针对某一箱区当天进口箱情况，本文决定在验证算法对不同规模算例有效性时选取集装箱规模分别为 20、40、60、80、100，表 5.3 为上述规模的算例在初始种群设为 40，经过 15 次重复实验 200 代时对应翻箱量的情况。

表 5.3 不同规模下集装箱的翻箱量

Tab. 5.3 The rehandles of different dimensions of containers

算例	集装箱规模	翻箱量			
		最大值	最小值	均值	平均翻箱率
算例 1	20	2	0	1	5.0%
算例 2	40	3	2	2.47	6.2%
算例 3	60	6	5	5.13	8.5%
算例 4	80	10	6	7.27	9.1%
算例 5	100	11	8	9	9.0%

通过表 5.3 可以看出，翻箱率都小于 10%，较目前场中平均 30%的翻箱率有较大改善。下表 5.4 为上述不同规模下算例其中一次的优化结果。

表 5.4 不同规模下的优化结果

Tab. 5.4 The optimization results of different dimensions of containers

20 个集 装 箱 箱 位 分 配 方 案	翻箱量	0
	卸船顺 序	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20
	提箱优 先级	17-13-5-8-10-20-11-5-21-7-16-3-7-8-6-11-6-11-2-5
	堆存位 置	2-4-5-2-3-6-4-5-1-3-6-2-4-6-3-1-6-1-5-4
	提箱顺 序	19-12-8-3-20-7-15-10-13-4-14-5-7-18-16-2-11-1-6-9
60 个集 装 箱 箱 位 分 配 方 案	翻箱量	5
	卸船顺 序	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25- 26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46- 47-48-49-50-51-52-53-54-55-56-57-58-59-60
	提箱优 先级	17-13-5-8-10-20-11-5-21-7-16-3-7-8-6-11-6-11-16-5-22-19-18-1-21-4- 17-9-24-1-23-4-6-1-5-9-15-13-2-22-6-4-12-8-23-7-21-16-14-24-24- 13-12-1-2-4-13-3-14-2
	堆存位 置	15-11-12-3-14-1-17-12-8-11-18-6-14-1-3-10-11-12-7-15-2-5-4-6-13- 17-4-10-16-12-10-15-14-15-14-18-5-6-3-2-10-7-2-18-9-17-9-4-9-16- 16-8-13-6-5-9-8-2-7-8
	提箱顺 序	30-34-54-24-39-55-60-12-58-32-56-26-42-20-35-8-3-15-17-33-41-10- 13-46-4-14-44-36-28-5-7-16-18-43-53-2-38-57-52-59-49-37-11-19-48- 1-27-23-22-6-9-25-47-40-21-31-45-51-50-29
80 个集 装 箱 箱 位 分 配 方 案	翻箱量	6
	卸船顺 序	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25- 26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46- 47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68- 69-70-71-72-73-74-75-76-77-78-79-80
	提箱优 先级	21-14-13-13-24-24-23-12-2-24-3-17-8-18-2-20-21-11-22-18-18-9-5-19- 7-11-19-20-21-8-5-22-14-15-15-11-7-12-23-1-10-19-1-2-12-3-4-14- 17-6-23-16-7-13-24-19-23-13-2-8-9-12-22-14-17-7-9-18-11-21-7-1-15- 4-4-10-15-5-6-3
	堆存位 置	16-11-12-17-18-22-19-16-15-6-14-2-12-19-14-5-7-17-20-20-22-7-3-5- 7-11-1-9-18-5-3-4-13-20-8-22-11-13-21-5-20-21-6-15-18-3-22-24-23- 4-3-23-16-8-14-9-10-9-6-9-24-8-10-6-19-14-18-1-8-10-11-12-21-16- 17-21-2-24-4-10

续表 5.4:

80个集装箱箱位分配方案	提箱顺序	40-72-43-44-9-59-15-11-80-46-47-74-75-31-23-78-79-50-25-71-37-66-53-30-60-13-22-61-67-41-76-18-26-36-69-38-45-62-8-3-4-58-54-2-33-64-48-34-35-73-77-52-12-49-65-14-20-68-21-68-24-27-42-56-16-28-1-17-29-70-19-32-63-7-39-51-57-5-6-10-55
	翻箱量	8
100个集装箱箱位分配方案	卸船顺序	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100
	提箱优先级	13-18-17-14-6-24-22-16-17-5-21-13-23-11-7-9-15-12-4-23-24-17-10-18-2-22-7-21-17-1-6-8-20-15-6-4-11-10-8-15-12-18-13-10-5-4-2-2-14-24-13-16-21-7-14-11-3-11-3-21-7-12-17-10-24-18-8-12-10-11-23-1-18-21-16-23-21-7-23-7-4-9-6-15-22-22-20-10-2-1-18-8-19-9-21-21-21-16-10-8
	堆存位置	26-24-5-13-28-4-9-29-5-28-23-3-11-5-25-15-19-22-16-11-17-25-13-20-28-18-15-8-30-15-11-27-4-2-25-19-5-10-10-29-9-12-12-24-13-24-19-16-23-21-29-20-1-27-30-11-27-18-23-4-12-22-4-18-6-1-26-3-30-22-7-19-6-8-1-17-8-1-7-16-24-30-23-8-21-21-14-3-7-22-13-2-14-21-10-12-17-14-29-10
	提箱顺序	72-90-30-25-47-89-48-57-59-19-36-81-46-10-45-5-35-83-31-27-54-78-80-15-61-32-92-100-39-67-16-82-94-23-38-44-64-69-88-99-37-14-56-58-70-41-62-18-68-1-12-43-51-4-49-55-17-34-40-84-8-52-75-98-9-3-22-29-63-2-24-42-66-73-91-93-87-33-11-60-53-77-74-28-95-96-97-7-26-86-85-20-13-79-71-76-6-21-50-65

由算例结果可知，在集装箱的卸船顺序以及货主提箱时间已知的情况下，由遗传算法确定的箱位分配方案能够大大减少提箱时的翻箱量，提高了堆场的作业效率，同时算例的结果也证明了本文模型算法的有效性。

5.4 本章小结

本章以一个具体的算例验证了算法的可行性与有效性，同时验证了算法在不同规模下的实用性，根据具体算例给出的箱位分配方案以及翻箱方案为以后堆场在制定相关决策时提供了一定的参考。

第6章 总结与展望

6.1 全文总结

集装箱后方堆场箱位分配问题是码头堆场管理的一个重要环节,然而在实际卸船时,堆场相关管理人员还主要依靠个人经验,较多堆场采用的堆存的方案仍是不同货主不同提单的进口箱分开堆放,这不仅浪费了空间资源,也没能充分发挥港口机械的效能。本文综合分析了国内外对集装箱箱位分配研究的现状,针对目前存在的实际问题,利用遗传算法就集装箱码头进口箱箱位分配提出优化方案。纵观全文,主要完成了以下工作:

(1) 本文通过对集装箱码头堆场中进口箱从卸船到被客户提走流程进行分析,总结造成翻箱的主要原因,对其中容易造成或导致翻箱的环节提出了相应的改进措施。

(2) 在计算堆场翻箱量时,本文摒弃以往求解翻箱量时的静态计算方法,提出了一种动态计算翻箱量的方法,每次只计算当前被提取目标箱的翻箱量,直至提取完所有集装箱。这种计算方法考虑到提箱过程中的翻箱对以后提取集装箱时的影响,更贴近实际中的翻箱量。

(3) 以提箱时最小翻箱量为目标,建立了后方堆场进口箱具体箱位分配问题的数学模型。本文在使用遗传算法解决问题时设计了合适的编码、遗传算子等,并根据模型编写了相应的求解程序,通过对具体算例的计算验证该优化算法能够达到满意的效果,为以后贝位分配问题的研究提供了基础,并为以后开发堆场管理软件提供了借鉴。

(4) 翻箱时落箱位置的确定。本文从二次翻箱的角度提出了一个翻箱时阻塞箱落箱位置的确定方案,并给出翻箱时具体落箱位置的确定规则。

6.2 未来展望

本文对集装箱后方堆场具体箱位分配问题进行了一定的研究,从源头减少了翻箱量,然而由于时间和其它方面的限制,仍然有许多方面有待进一步的深入研究和改进完善:

(1) 本文仅对一批进口箱进行箱位分配,堆场初始状态比较简单,未考虑后

方堆场有卸有提的情况，在以后的研究中要考虑堆场更复杂操作流程，给出一个滚动的堆存方案，使之更贴近码头进口箱箱位分配的实际情况。同时文中的卸船顺序已知，在后续的研究中可以探讨一下卸船顺序不确定条件下的集装箱在堆场中的堆存问题。

(2) 用遗传算法解决此类问题时，要进一步改善算法的性能，寻求更好的编码选择变异策略。许多研究人员也尝试采用其它现代优化技术来解决具体箱位的分配问题，孰优孰劣，尚需时间验证，因此寻找一种更优的优化算法将是一个长期的目标。

(3) 随着堆场环节上越来越普遍的信息化，未来对集装箱信息掌握的会更多，要利用好获取的信息，同时由于堆场中某些不确定因素的存在，算法要增强对某些突发状况的抵抗能力。

参考文献

- [1] Kim K H. Evaluation of the number of rehandles in container yards[J]. Computers & Industrial Engineering, 1997, 32(4): 701-711.
- [2] Kim K H, Park Y M, Ryu K R. Deriving decision rules to locate export containers in container yards[J]. European Journal of Operational Research, 2000, 124(1): 89-101.
- [3] Roll Y, Rosenblatt M J. Random versus grouped storage policies and their effect on warehouse capacity[J]. Material Flow, 1983, 1(3): 199-205.
- [4] Taleb-Ibrahimi M, de Castilho B, Daganzo C F. Storage space vs handling work in container terminals[J]. Transportation Research Part B: Methodological, 1993, 27(1): 13-32.
- [5] Zhang C, Liu J, Wan Y, et al. Storage space allocation in container terminals[J]. Transportation Research Part B: Methodological, 2003, 37(10): 883-903.
- [6] 王斌. 集装箱码头堆场的一种动态随机堆存方法[J]. 系统工程理论与实践, 2007, 27(4): 147-153.
- [7] 刘艳, 周鹏飞, 康海贵. 集装箱堆场箱位分配模糊优化研究[J]. 水运工程, 2009 (11): 1-5.
- [8] 陶经辉, 汪敏. 基于混堆模式的集装箱堆场区段分配[J]. 系统工程理论与实践, 2009 (008): 185-192.
- [9] Zhang C. Resource planning in container storage yard[D]. Hong Kong University of Science and Technology, 2000.
- [10] Zhang C, Wan Y, Liu J, et al. Dynamic crane deployment in container storage yards[J]. Transportation Research Part B: Methodological, 2002, 36(6): 537-555.
- [11] 冯美玲, 岳文英, 孙俊清. 进出口集装箱在堆场中堆存空间分配问题的研究[J]. 中国自动化学会控制理论专业委员会 D 卷, 2011.
- [12] 金海龙. 紧凑型集装箱码头重箱堆场堆存策略应用研究[D]. 上海交通大学, 2010.
- [13] 周留井. 基于遗传算法的进口箱堆场 BAY 位分配优化模型研究[D]. 广州: 中山大学, 2008.

- [14]杨超, 陆志强. 基于遗传算法的出口集装箱送箱时间窗优化[J]. 计算机应用研究, 2013, 30(6): 1643-1646.
- [15]魏航. 集装箱码头堆场取箱时间窗优化研究[J]. 管理科学学报, 2011, 14(9): 21-36.
- [16]孙俊清, 岳文英, 杨鹏. 基于 GA 的集装箱在堆场中堆存位置的决策[J]. 中国自动化学会控制理论专业委员会 D 卷, 2011.
- [17]郝聚民, 纪卓尚. 混合顺序作业堆场 BAY 优化模型[J]. 大连理工大学学报, 2000, 40(1): 102-105.
- [18]陈庆伟, 王继荣. 集装箱堆场出口箱堆存模型及其算法[J]. 物流科技, 2007, 30(7): 106-108.
- [19]谢尘, 何军良, 茆道方. 基于混堆模式的集装箱码头出口箱进场选位策略[J]. 上海海事大学学报, 2009, 29(4): 70-75.
- [20]周鹏飞. 面向不确定环境的集装箱码头优化调度研究 [D]. 大连理工大学博士论文, 2005.
- [21]徐亚, 陈秋双, 龙磊, 等. 集装箱倒箱问题的启发式算法研究[J]. 系统仿真学报, 2008, 20(14): 3666-3669.
- [22] Lee Y, Hsu N Y. An optimization model for the container pre-marshalling problem[J]. Computers & Operations Research, 2007, 34(11): 3295-3313.
- [23] Lee Y, Chao S L. A neighborhood search heuristic for pre-marshalling export containers[J]. European Journal of Operational Research, 2009, 196(2): 468-475.
- [24] Laik N, Hadjiconstantinou E. Container assignment and yard crane deployment in a container terminal: a case study[J]. Maritime Economics & Logistics, 2008, 10(1): 90-107.
- [25]易正俊, 江静, 胡勇. 堆场集装箱翻箱的 PCNN 优化控制算法[J]. 自动化学报, 2011.
- [26]王洪雪, 雷黎黎. 集装箱堆场箱位最优分配[J]. 交通科技与经济, 2013, 15(1): 50-53.
- [27]董琳, 刘庆敏, 王超, 等. 集装箱翻箱问题的模型分析及算法[J]. 经济数学, 2006, 23(2): 181-186.
- [28]易正俊, 李保顺, 李新强. 集装箱堆场倒箱博弈启发式优化算法[J]. 上海海事大学学报, 2010, 31(003): 47-51.

- [29] 李斌. 基于动态规划的贝位内集装箱翻箱优化[D]. 大连海事大学, 2011.
- [30] 缪立新. 集装箱装船顺序优化模型及算法研究 [C][J]. 第三届中国物流学术年会论文集, 2004.
- [31] 王志明. 集装箱码头后方堆场箱位分配优化研究[D]. 重庆大学, 2010.
- [32] 陈智. 中小码头堆场翻箱问题优化模型及优化设计[D]. 武汉理工大学, 2010.
- [33] 刘英. 遗传算法中适应度函数的研究[J]. 兰州工业高等专科学校学报, 2006, 13(3): 1-4.
- [34] 韩万林, 张幼蒂. 遗传算法的改进[J]. 中国矿业大学学报, 2000, 29(1): 102-105.
- [35] 黄贞程. 如何减少堆场进口重箱盘箱[J]. 集装箱化, 2009, 20(8): 17-18.
- [36] 钟慧玲, 欧仙群, 张冠湘, 等. 随机集港的出口箱堆存策略设计与仿真[J]. 上海海事大学学报, 2012, 33(1): 26-31.
- [37] Kim K H, Hong G P. A heuristic rule for relocating blocks[J]. Computers & Operations Research, 2006, 33(4): 940-954.
- [38] 赵宁, 宓为建, 邓钟. 集装箱码头卸船箱进场选位算法研究[J]. 中国工程机械学报, 2010 (1): 17-23.
- [39] Ottjes J A, Veeke H P M, Duinkerken M B, et al. Simulation of a multiterminal system for container handling[M]//Container terminals and cargo systems. Springer Berlin Heidelberg, 2007: 15-36.
- [40] 李丕安. 堆场进口箱翻箱落位优选干扰管理模型与算法[D]. 大连理工大学, 2013.
- [41] Yang J H, Kim K H. A grouped storage method for minimizing relocations in block stacking systems[J]. Journal of Intelligent Manufacturing, 2006, 17(4): 453-463.
- [42] Zhang C, Chen W, Shi L, et al. A note on deriving decision rules to locate export containers in container yards[J]. European Journal of Operational Research, 2010, 205(2): 483-485.
- [43] Bortfeldt A, Gehring H, Mack D. A parallel tabu search algorithm for solving the container loading problem[J]. Parallel Computing, 2003, 29(5): 641-662.
- [44] 周鹏飞, 李丕安. 集装箱堆场不确定提箱次序与卸船箱位分配[J]. 哈尔滨工程大学学报, 2013, 34(9): 1119-1123.
- [45] 周鹏飞, 李丕安. 不确定条件下集装箱堆场出口箱具体箱位优选[J]. 工业工程, 2013 (1): 25-30.

- [46] Hartmann S. Generating scenarios for simulation and optimization of container terminal logistics[J]. OR Spectrum, 2004, 26(2): 171-192.
- [47] Kim K H, Kim D Y. Group storage methods at container port terminals[J]. The American Society of Mechanical Engineers, 75th Anniversary Commemorative Volume, MH, 1994, 2: 15-20.
- [48] Dekker R, Voogd P, van Asperen E. Advanced methods for container stacking[M]//Container terminals and cargo systems. Springer Berlin Heidelberg, 2007: 131-154.
- [49] Lee B K, Kim K H. Optimizing the block size in container yards[J]. Transportation Research Part E: Logistics and Transportation Review, 2010, 46(1): 120-135.
- [50] Lee D H, Cao Z, Meng Q. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm[J]. International Journal of Production Economics, 2007, 107(1): 115-124.
- [51] Caserta M, Schwarze S, Voß S. A mathematical formulation and complexity considerations for the blocks relocation problem[J]. European Journal of Operational Research, 2012, 219(1): 96-104.
- [52] 杨淑芹, 张运杰. 集装箱堆场问题的一个数学模型及其算法[J]. 大连海事大学学报: 自然科学版, 2002, 28(B08): 115-117.
- [53] 朱明华, 范秀敏, 程免翀, 等. 集装箱装船顺序问题的启发式算法研究[J]. 2010.
- [54] 王孟昌. 集装箱码头堆场箱位动态分配优化策略研究[D]. 武汉理工大学, 2007.
- [55] 王莉莉, 于红. 集装箱装船顺序优化模型及遗传算法[J]. 计算机工程与应用, 2008, 44(15): 234.
- [56] 黎明, 翟金刚. 基于粒子群算法的出口集装箱堆场箱位分配研究[J]. 水运工程, 2012 (11): 42-46.
- [57] 孙环环. 基于船舶配载图的集装箱装船顺序优化模型与算法研究[D]. 大连海事大学, 2012.
- [58] MATLAB 遗传算法工具箱及应用[M]. 西安电子科技大学出版社, 2005.
- [59] 曹道友, 程家兴. 基于改进的选择算子和交叉算子的遗传算法[J]. 计算机技术与发展, 2010, 20(2): 44-47.

- [59]李鹏, 董聪. 基于实数编码的广义遗传算法及其在优化问题中的应用[J]. 控制与决策, 2002, 17(4): 487-490.
- [60]周育人, 王勇. 一种有效的实数编码遗传算法[J]. 武汉大学学报: 理学版, 2003, 49(1): 39-43.

附录 A 100 个进口箱箱位分配主程序代码

```

clc;clear
T1=clock;
total=100;%卸箱总数
generation=80;%种群数
total_stack=30;
for i=1:80
    a=[[1:30],[1:30],[1:30],[1:30]];
    b=a(randperm(length(a)));
    chrom(i,:)=b(1:100);
end
chrom1=chrom(41:80,:);
chrom2=[];
for i=1:20
    x=randi([1,total],1,1);
    y1=0;
    y2=0;
    while y1==y2
        y1=randi([1,40],1,1);
        y2=randi([1,40],1,1);
    end
    z1=[chrom1(y1,1:x),chrom1(y2,x+1:total)];
    z2=[chrom1(y2,1:x),chrom1(y1,x+1:total)];
    d1=zeros(1,total_stack);
    d2=zeros(1,total_stack);
    for j=1:total_stack
        c1=find(z1==j);
        c2=find(z2==j);
        d1(j)=4-size(c1,2);% 正数表示栈中有空位，负数表示栈已超高
        d2(j)=4-size(c2,2);
    end
    for j=1:total_stack
        if d1(j)<0
            c1=find(z1==j);% 找出超高栈中的所有箱
            o1=randperm(size(c1,2));%集装箱随机排序
            count=1;
            for k=1:(-d1(j))%超高箱的数量
                p1=find(d1>0);% 找到有空箱位的栈
                q1=randi([1,size(p1,2)],1,1);%随机选择一个栈
                z1(c1(o1(count)))=p1(q1);
                d1(p1(q1))=d1(p1(q1))-1;
                count=count+1;
            end
        end
    end
end

```

```

        d1(j)=0;
    end
    if d2(j)<0
        c2=find(z2==j);%找出超高栈中所有箱
        o2=randperm(size(c2,2));%箱子随机排序
        count=1;
        for k=1:(-d2(j))%超高箱数量
            p2=find(d2>0);% 找到有空箱位的栈
            q2=randi([1,size(p2,2)],1,1); %随机选取一个栈
            z2(c2(o2(count)))=p2(q2);
            d2(p2(q2))=d2(p2(q2))-1;
            count=count+1;
        end
        d1(j)=0;
    end
    end
    chrom2=[chrom2;z1;z2];
end
chrom=[chrom1;chrom2];
time=[13    18    17    14    6    24    22    16    17    5    21    13    23
11    7    9    15    12    4    23    24    17    10    18    2    22    7    21
17    1    6    8    20    15    6    4    11    10    8    15    12    18    13
10    5    4    2    2    14    24    13    16    21    7    14    11    3    11
3    21    7    12    17    10    24    18    8    12    10    11    23    1    18
21    16    23    21    7    23    7    4    9    6    15    22    22    20    10
2    1    18    8    19    9    21    21    21    16    10    8];
[over,A,sequence]=stowage(generation,total,total_stack,chrom,time);
iter=200;%迭代次数
out=randi([1,total-1],1,1);
gvalue=[];
com=[10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,180,190,200,210,220,230,240,25
0,260,270,280,290,300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,450,460,470,480,4
90,500,510,520,530,540,550,560,570,580,590,600,610,620,630,640,650,660,670,680,690,700,710,720,
730,740,750,760,770,780,790,800];
for ii=1:iter
    compare=intersect(ii,com);
    if isempty(compare)==0
        [over,A,sequence]=stowage(generation,total,total_stack,chrom,time);
        media=[over',chrom];
        media=sortrows(media,1);
        chrom1=media(1:40,2:total+1);
        chrom2=[];
    end
end
for i=1:20
    x=randi([1,total],1,1);

```

```

y1=0;
y2=0;
while y1==y2
y1=randi([1,80],1,1);
y2=randi([1,80],1,1);
end
z1=[chrom(y1,1:x),chrom(y2,x+1:total)];
z2=[chrom(y2,1:x),chrom(y1,x+1:total)];
d1=zeros(1,total_stack);
d2=zeros(1,total_stack);
for j=1:total_stack
    c1=find(z1==j);
    c2=find(z2==j);
    d1(j)=4-size(c1,2);% 正数表示栈中有空位，负数表示栈已超高
    d2(j)=4-size(c2,2);
end
for j=1:total_stack
    if d1(j)<0
        c1=find(z1==j);% 找出超高栈中的所有箱
        o1=randperm(size(c1,2));%集装箱随机排序
        count=1;
        for k=1:(-d1(j))%超高箱数量
            p1=find(d1>0);% 找到有空箱位的栈
            q1=randi([1,size(p1,2)],1,1);%随机选取一个栈
            z1(c1(o1(count)))=p1(q1);
            d1(p1(q1))=d1(p1(q1))-1;
            count=count+1;
        end
        d1(j)=0;
    end
    if d2(j)<0
        c2=find(z2==j);% 找出超高栈中的所有箱
        o2=randperm(size(c2,2));% 箱子随机排序
        count=1;
        for k=1:(-d2(j))
            p2=find(d2>0);
            q2=randi([1,size(p2,2)],1,1);
            z2(c2(o2(count)))=p2(q2);
            d2(p2(q2))=d2(p2(q2))-1;
            count=count+1;
        end
        d2(j)=0;
    end
end
end

```

```

    chrom2=[chrom2;z1;z2];
end
chrom=[chrom1;chrom2];
    end
    AB=chrom;
if out==total
    out=randi([1,total-1],1,1);
end
if AB(out)~=AB(out+1)
M=AB(:,out);
AB(:,out)=AB(:,out+1);
AB(:,out+1)=M;
[over1,A,sequence]=stowage(generation,total,total_stack,AB,time);
for i=1:generation
    if over(i)>over1(i)
chrom(i,:)=AB(i,:);
        else if over(i)==over1(i)
            Part=A(:,(5*i-4):5*i);
            stack1=Part(out,4);
            P1=Part(out,2);%换后的优先级
            stack2=Part(out+1,4);
            P2=Part(out+1,2);
            Part(out,:)=[];
            Part(out,:)=[];
            t1=find(Part(:,4)==stack1);
            t2=find(Part(:,4)==stack2);
            tt1=(Part(t1,2)-P1)*(Part(t1,2)-P1);
            tt2=(Part(t2,2)-P2)*(Part(t2,2)-P2);
            d1=find(tt1==1);
            d2=find(tt2==1);
            if isempty(d1)+isempty(d2)==0
                chrom(i,:)=AB(i,:);
            end
        end
    end
end
end
end
out=out+1;
[over,A,sequence]=stowage(generation,total,total_stack,chrom,time);
Bestvalue=min(over);
gvalue=[gvalue,Bestvalue];
end
minimum=find(over==min(over));
minimum(1)%翻箱数最小的种群编号

```

```
SE=sequence(total*(minimun(1)-1)+1:total*minimun(1))%提箱顺序
plot(gvalue,'b',LineWidth,2);
title('Best Solution History');
set(gca,'XLim',[0 iter+10],'YLim',[0 1.1*max([1 gvalue])]);
T2=clock;
etime(T2,T1)
```

附录 B 翻箱量子函数代码

```

function [over,A,sequence]= stowage(generation,total,total_stack,Chrom,time)
%UNTITLED Summary of this function goes here
[priority,order]=sort(time');
num_p=unique(priority');
A=zeros(total,5*generation);
for i=1:generation
    B=zeros(1,total_stack);
    for j=1:total
        A(j,(i-1)*5+1)=j;%集装箱序号
        A(j,(i-1)*5+2)=time(j);%提箱优先级
        A(j,(i-1)*5+4)=Chrom(i,j);%所在栈
        B(1,Chrom(i,j))=B(1,Chrom(i,j))+1;
        A(j,i*5)=B(1,Chrom(i,j));
        if Chrom(i,j)<=6
            A(j,(i-1)*5+3)=1;%所在堆场贝位
        else if Chrom(i,j)<=12
            A(j,(i-1)*5+3)=2;
        else if Chrom(i,j)<=18
            A(j,(i-1)*5+3)=3;
        else if Chrom(i,j)<=24
            A(j,(i-1)*5+3)=4;
        else if Chrom(i,j)<=30
            A(j,(i-1)*5+3)=5;
        end
    end
end
end
end
end
end
end
A;%初始种群(集装箱序号+卸船顺序+提箱优先级+栈)
M=6*ones(1,total_stack);
sequence=[];
over=[];
for i =1:generation
    C=A(:,(i-1)*5+1):(i*5);
    E=zeros(1,total_stack);
    for m=1:total_stack
        D=find(C(:,4)==m);
        if isempty(D)==1
            E(m)=0;
        else

```



```

E(m)=max(C(D,5));
end
end
CC=sortrows(C,2);%按照货主提箱优先级排序，对应着priority下的顺序，对应着level顺序
overl=0;
for j=1:size(num_p,2)
    column=find(priority==num_p(j));
    if size(column,1)>1
        ad=1;
        SN=CC(column,1);
        while isempty(column)==0
            start=10;
            judge=0;
            for k=1:size(column,1)
                if CC(column(k),5)==E(CC(column(k),4))
                    judge=1;
                    sequence=[sequence,SN(k)];
                    t=find(C(:,1)==SN(k));
                    stack=C(t,4);
                    E(stack)=E(stack)-1;
                    C(t,:)=[];
                    column(k)=[];
                    SN(k)=[];
                    break;%找到可以直接提取的集装箱，跳出for循环
                end
            end
            ss=E(CC(column(k),4))-CC(column(k),5);
            if ss<start
                start=ss;%翻箱次数
                container=SN(k);%确定待提箱（判断标准：压箱数最小的先提）
                stack=CC(column(k),4);
            end
        end
    end
    if judge==0%没有找到可以直接提取的集装箱
        if stack<=6
            P=setdiff([1,2,3,4,5,6],stack);
        else if stack<=12
            P=setdiff([7,8,9,10,11,12],stack);
        else if stack<=18
            P=setdiff([13,14,15,16,17,18],stack);
        else if stack<=24
            P=setdiff([19,20,21,22,23,24],stack);
        else if stack<=30
            P=setdiff([25,26,27,28,29,30],stack);
        end
    end
end

```



```

        stack1=P(k);
        low=E(P(k));
    end
    end
    end
    end
    E(stack1)=E(stack1)+1;
    E(stack)=E(stack)-1;
    CC(st1,4)=stack1;
    CC(st1,5)=E(stack1);
    C(st,4)=stack1;
    C(st,5)=E(stack1);
    end
    over1=over1+start;
end
end

end
while isempty(column)==0
    if size(column,1)==1&&CC(column,5)==E(CC(column,4))
        E(CC(column,4))=E(CC(column,4))-1;
        sequence=[sequence,CC(column,1)];%要提箱刚好在最上层则直接提
        row=find(C(:,1)==CC(column,1));
        C(row,:)=[];
        column=[];
    else if
        size(column,1)==1&&CC(column,5)<=E(CC(column,4))%该箱不位于某栈的最上
层，同时仅有一个箱当前可提
        start=E(CC(column,4))-CC(column,5);%需要翻箱次数
        stack=CC(column,4);
        if stack<=6
            P=setdiff([1,2,3,4,5,6],stack);
        else if stack<=12
            P=setdiff([7,8,9,10,11,12],stack);
        else if stack<=18
            P=setdiff([13,14,15,16,17,18],stack);
        else if stack<=24
            P=setdiff([19,20,21,22,23,24],stack);
        else if stack<=30
            P=setdiff([25,26,27,28,29,30],stack);
        end
    end
end
end
end
end

```

```

end
PP=P;
for p=1:start
    EM=M-E;
    Q=[];
    qq=size(PP,2);
    for td=1:qq
        if EM(PP(td))>0
            Q=[Q,PP(td)];
        end
    end
    P=Q;
    low=100;
    stack_row=find(C(:,4)==stack);
    tier=find(C(:,5)==E(stack));
    st=intersect(stack_row,tier);% 所倒箱在C中的行数
    st1=find(CC(:,1)==C(st,1));% 所倒箱在CC中的行数
    for k=1:size(P,2)
        r=find(C(:,4)==P(k));
        if isempty(r)==1
            stack1=P(k);
            break;
        else if C(st,2)<=min(C(r,2))
            stack1=P(k);
            break;
        else
            if E(P(k))<low
                stack1=P(k);
                low=E(P(k));
            end
        end
    end
    end
    E(stack1)=E(stack1)+1;
    E(stack)=E(stack)-1;
    CC(st1,4)=stack1;
    CC(st1,5)=E(stack1);
    C(st,4)=stack1;
    C(st,5)=E(stack1);
end
over1=over1+start;
end
end
end

```

```
    end
    over=[over,over1];
    end
end
```

致 谢

本论文是在我的导师计明军教授悉心指导下完成的，论文从选题、初稿修改到最终定稿的整个过程中，计老师付出了很多的心血，提出了很多的有益意见。计老师学术态度严谨、教学风格求实、深厚的理论功底使我在论文的研究期间不仅在学术上受益匪浅，也为我在今后的工作学习中树立了榜样。近两年来，计老师不仅在学业上给我以精心指导，同时还在思想、生活上给我以无微不至的关怀，在此谨向计老师致以诚挚的谢意和崇高的敬意。

本论文在写作过程中王清斌老师、杨永志老师也提出了许多意见，此外实验室的同学祝慧灵等也对论文的定稿起到了重要作用，在此一并感谢他们，也感谢他们对我学业及生活中的各种帮助。

此外，感谢家人对我读研的支持，感谢交通运输管理学院的各位老师的谆谆教导，感谢班级和寝室的各位同学对我的各种帮助和关怀，感谢各位专家老师在百忙之中审阅本文并提出宝贵意见，在这里请一并接受我诚挚的谢意！