

分类号: _____

密 级: _____公开_____

U D C: _____

单位代码: _____11646_____

宁波大学

硕士专业学位论文

论文题目: 全航线集装箱船与堆场混合配载优化

学 号: _____1411084976_____

姓 名: _____蔡兴武_____

专业学位类别: _____工程硕士_____

专业学位领域: _____船舶与海洋工程_____

学 院: _____海运学院_____

指 导 教 师: _____胡燕海 教授_____

合 作 教 师: _____彭捷 高级工程师_____

论文提交日期: 2017 年 4 月 2 日

A Thesis Submitted to Ningbo University for the Master's Degree

**Research on The optimization of full routes ship and
container yard mixed stowage**

Candidate: Cai xing-wu

Supervisor: professor Hu yan-hai

Faculty of Maritime and transportation
Ningbo University
Ningbo 315211, Zhejiang P.R.CHINA

April 15, 2017

独 创 性 声 明

本人郑重声明：所呈交的论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得宁波大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

签名：_____ 日期：_____

关于论文使用授权的声明

本人完全了解宁波大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或者部分内容，可以采用影印、缩印或其它复制手段保存论文。

（保密的论文在解密后应遵循此规定）

签名：_____ 导师签名：_____ 日期：_____

全航线集装箱船与堆场混合配载优化

摘 要

本文在调研集装箱船配载问题国内外研究现状的基础上,提出了面向全航线的集装箱船与堆场混合配载的方法,建立了全航线集装箱船与堆场整体倒箱量最小、出口箱区取箱时场桥贝位间往复移动次数最少的数学模型,运用遗传算法对模型进行求解,通过 Matlab 进行数值模拟。

本文依据堆场出口箱区随机的集装箱堆存状态,通过变换取箱顺序,得到船舶中集装箱的配载,使得堆场取箱的倒箱量和船舶倒箱量之和最少的时候,场桥的贝位间往复移动次数最少化。通过与传统配载模式的对比,验证了全航线集装箱船与堆场混合配载方法对于减少倒箱,提高装卸效率有着积极意义。论文主要内容展开如下:

首先,介绍集装箱码头布局中的堆场,包括定义,堆场主要装卸设备,集装箱在堆场的堆存,堆场倒箱。定义了集装箱配积载及其原则,介绍了船舶坐标表示方法、船舶倒箱,为后续设计提出的混合配载奠定基础。

其次,提出全航线配载时假设条件和变量设计,以全航线装卸时堆场及船配载所产生的倒箱量最小和出口箱区场桥贝位间来回大距离移动次数最小为目标建立数学模型。

再次,用遗传算法进行求解模型。以倒箱量和场桥贝位间往复移动次数作约束,运用扫描法取堆场堆箱;配载时,设计编码满足船舶舱盖下不产生倒箱,舱盖上产生混 bay 来预调船舶安全性能。

最后,将混合配载方法应用于宁波港吉码头经营有限公司,通过实际数据计算验证了本文提出的全航线集装箱船与堆场混合配载方法在减少倒箱方面有着积极意义。

关键词: 堆场, 全航线, 配载, 倒箱, 场桥移动, 遗传算法.

The optimization of full routes ship and container yard mixed stowage

Abstract

Based on the study of the researching the domestic and foreign research status of container ship loading problem, the container yard and ship mixed stowage method was proposed. With the objective of minimizing the total shift of container yard and ship and the total number of container yard cranes reciprocating moved between two bay, an optimization model was established. Genetic algorithm is used to optimize the model, Numerical simulation by Matlab.

Based on the random set of export container yard, through the transformation from loading sequence, get the container stowage make total shift of container yard and ship and total container yard cranes moved back and forth between two bay more less. Compared with the traditional stowage model, the new model are confirmed of great significance to improve the shift less, improve the efficiency of equipment. The main contents of this paper are introduced as follow:

Firstly, introduce the layout of container terminal, including the definition, main yard handling equipment in container yard, the yard stockpiling, container yard stowage. Introduce the definition and principles of the ship stowage, the ship coordinates method, ship shift, to lay the foundation for the subsequent design of the mixed loading is proposed.

Secondly, put forward the assumption and variable design of the whole route stowage, With the objective of minimizing the total shift of container yard and ship and the total container yard cranes moved back and forth between two bay, an optimization model was established. Using genetic algorithm to solve.

Thirdly, using genetic algorithm to solve the new model. Made shift of container yard and ship and the total number of container yard cranes reciprocating moved between two bay as a constraint, used the scanning method to loading; As a stowage plan, design code to meet the ship under the hatch does not produce shift and hatch on the mixed bay to preadjust the ship's safety performance.

Finally, the new method is applied to Ningbo Gangji Terminal Company Limited of Ningbo Beilun Harbor. By the real data calculation verified that the method of mixed stowage of ship and container yard is of positive significance in reducing the shift.

Key words: Container yard, Full route , Stowage, Shift, Container yard cranes movement , Genetic algorithm.

目 录

摘 要.....	I
Abstract	II
目 录.....	III
1 绪论.....	- 1 -
1.1 研究背景及意义.....	- 1 -
1.2 国内外研究现状.....	- 5 -
1.2.1 装船过程中集装箱配载研究	- 5 -
1.2.2 关于堆场堆存优化研究.....	- 6 -
1.2.3 关于遗传算法的研究	- 7 -
1.3 本文研究内容	- 8 -
1.4 论文通篇结构	- 8 -
2 集装箱配载涉及内容概述	- 10 -
2.1 集装箱堆场相关概述	- 10 -
2.1.1 堆场定义	- 10 -
2.1.2 堆场装卸设备.....	- 10 -
2.1.3 集装箱堆场堆存状态	- 11 -
2.1.4 堆场倒箱	- 12 -
2.2 集装箱船相关概述	- 13 -
2.2.1 集装箱船配载的概念	- 13 -
2.2.2 集装箱船配积载原则	- 14 -
2.2.3 集装箱船坐标系	- 16 -
2.2.4 集装箱在船舶上的布局.....	- 17 -
2.2.5 集装箱在船上倒箱	- 17 -
2.3 本章小结	- 18 -
3 全航线集装箱船与堆场混合配载模型建立.....	- 19 -
3.1 问题的提出	- 19 -
3.2 基本内容	- 19 -
3.2.1 运输矩阵及同类箱组定义	- 19 -
3.2.2 堆场出口箱区矩阵级场桥多余移动次数定义	- 20 -
3.2.3 权重 ¹ 的确定	- 22 -

3.2.4 船舶箱位的表示方法	23 -
3.2.5 船舶分区表示方法	23 -
3.3 混合配载模型建立	24 -
3.3.1 模型假设	24 -
3.3.2 参数和变量设计	24 -
3.3.3 优化目标及约束	26 -
3.4 本章小结	27 -
4 优化算法.....	28 -
4.1 遗传算法流程	28 -
4.2 初始解求解	29 -
4.3 染色体编码	30 -
4.3.1 基于出口箱区混堆的染色体设计	30 -
4.3.2 堆场取箱顺序下的船舶配载设计	31 -
4.4 适应度函数	32 -
4.5 选择操作	32 -
4.6 遗传算子	33 -
4.6.1 交叉操作	33 -
4.6.2 变异操作	33 -
4.7 数值模拟	33 -
4.8 本章小结	38 -
5 工程应用.....	39 -
5.1 SPARCE 系统简介	39 -
5.2 实例数据	40 -
6 结论与展望	44 -
6.1 研究结论与创新	44 -
6.2 研究展望	44 -
参考文献	45 -

1 绪论

1.1 研究背景及意义

集装箱运输是将货物处理后组装在集装箱（container），以集装箱为最小单位，通过交通工具运输集装箱的一种运输方式。其起源于英国，后续传到美、德、法等欧美国家，并得到大力发展。集装箱运输到港后，运用大型机械装载集装箱或卸载集装箱，从而更高效率和更高效益的实现两地间货物的运输。集装箱运输可根据选择不同的运输路线来选取运输工具，如用船舶运输、用火车运输、用集卡运输等，在运输过程中，若需要中途换运输方式，可以直接取整个集装箱进行操作，根本不需要将集装箱内部的货物取出。以集装箱作为运输单元一方面能更利于运输的机械化，提高操作效率，减少工作时间。另一方面，机械化操作可以减少人为操作造成的货物损失等，集装箱运输已经取代了传统的运输方式，成为各国运输现代化的标志。

集装箱按照不同的需求可以分成不同类型，按货物种类分集装箱可以分为干货集装箱（DC）、散货集装箱（BC）、液体货集装箱（LC）、冷藏箱集装箱（CC）、特种专用集装箱；按用途分有冷冻集装箱（RF）、挂衣集装箱（DHC）、开顶集装箱（OT）、框架集装箱（FR）、罐式集装箱（TK）、冷藏集装箱、平台集装箱（PC）、通风集装箱（VC）、保温集装箱（IC）。国际上通常使用的干货柜（DC）有：20ft 和 40ft。图 1.1 为干货柜集装箱（DC）的示例。

集装箱运输需要集装箱船、港口、集卡等大量的周转，这不仅需要大量的资本，更需要先进的管理技术和管理人才，一般的船公司根本负担不起，发展中国家由于各种限制和原因起步都较晚，只处于发展阶段，甚至处于起步阶段。我国海上物流起步也相对较晚，但改革开放三十年来，随着国家的重视，集装箱运输还是得到了迅猛的发展，甚至在 2008 年经金融危机到现在船市行情持续低迷情况下，2015 年，我国规模以上港口完成货物吞吐量 114.3 亿 t，同比增长 1.6%，绝对增量达到近 1000 万 TEU。从吞吐量增速的缓慢上升的趋势看，集装箱运输前景依然乐观。



图 1.1 干货柜集装箱

Figure 1.1 Dry container

从上海国际航运中心收集的 2017 年第一季度全球港口发展报告可以了解到最近行情。本季度的全球经济增速较上季度依然是缓行趋势，复苏到 2008 前繁荣景象的进程仍呈现疲弱状态，未来集装箱港口发展的基调依旧是低速稳增型。从表 1.1 可以看出，我国集装箱吞吐量仍稳固前行，因此研究集装箱相关问题对我国经济发展的推进有着重要意义。

表 1.1 2017 年第一季度吞吐量全球港口排名

Table1.1 Container throughput of the first quarter of 2017global port rankings

2017年1-3月全球各大港口集装箱吞吐量及相对指标一览表				
单位：万标准箱、%				
	3月吞吐量	同比增长	自年初累计吞吐量	累计同比增长
上海	340	13.3	935.32	9.5
新加坡	270	8	760	2.7
宁波-舟山港	200.24	14.06	588.06	9.1
深圳	190.05	5.81	561.37	1.03
香港	177	17.1	486.7	12.6

目前我国几个比较有名的港口包括：宁波-舟山港、上海港等。这些港口通过各种智能软件已经实现了码头作业和管理的自动化。

港口方面，集装箱可以在码头周转，集装箱可以在码头交接货物，集

装箱还可以堆放在码头，因此码头是集装箱缓冲地，在整个运输体系中码头占有很足的分量。由码头功能决定了集装箱码头建设面积都比较大、装卸设备多、设备齐全等特点，是海铁联运、公路运输的重要中转站。港口码头的发展也是体现集装箱运输的一重要标志。如何改进装卸模式，如何合理规划堆场布局，如何分配岸桥调度，如何科学分配集卡装卸，如何合理化船舶配载等等，这些问题的解决都能提高集装箱装卸效率，成为了如何提高码头效率的问题。目前国内外的专家、工作人员也在积极研究这些问题的解决方法。

除了上文提到的港口码头扮演的角色和作用，另外一影响集装箱运输的重要环节是集装箱的配载。目前集装箱配载的问题应用于多个领域，已得到了长足进步，物流运输业就是其中一个应用实例。解决集装箱的配载问题将更合理利用船舶空间利用率、改善船舶停靠的时间效率从而直接影响行业的成本和收益。

集装箱配载货物是将准备出口的集装箱按造运输时满足船舶的各种要求，如要满足船舶稳性、保证船舶强度、达到吃水差等要求和按照卸载时满足港口码头的各种要求，如船舶的停靠、装卸效率、岸桥利用率等要求来制定的具体装载图。制定出的配载图只作为指导文件，具体情况，工作人员会根据工作经验来操作。

配积载的好可以大大的降低船舶的运输效率让运输成本降到最低。总的来说配积载主要体现在：

（1）配积载的好坏直接影响船舶装卸作业的工作效率。好的配载计划能显著提高装船效率，但配载问题不是简单的装船顺序的制定，它涉及到各船舶航线的协调问题、各岸桥的配合问题、以及其他各类问题，在制定配载计划过程中要充分考虑到以上各种因素。目前多以软件预配，实际配载多以经验为主；

（2）配积载的好坏直接影响船舶的航行安全。工作人员根据制作的配载图将集装箱装船后，集装箱的重量会影响船舶的稳性、强度、吃水差等。配载时装载的集装箱的重量应该不超过额定限制，要特别注意船舶的纵向强度，以避免船舶中拱的产生。同时，为保持船舶的各种安全性能，如稳性和浮态，工作人员可以利用船舶的箱位进行调整，也可以合理安排轻重箱的位置，以达到船舶左右前后的重量差在规定荷载内。在考虑到这些因素后，制作出的配载图才能保证船舶的安全航行；

（3）配积载的好坏能直接影响倒箱量的产生。在制作配载图时，若没有大局观，没有考虑全航线的配载，没有考虑各港的箱源情况，只考虑

当前港口的配载，会造成后续港口的互压，这样会产生大量倒箱，好的配载可以避免大量倒箱，保证岸桥工作量尽可能少，以提高经济效益。

然而再好的配载也会因为各种原因产生倒箱。

工作人员评价装船顺序优劣的可以直接根据倒箱量的多少来判断。考虑到集装箱船的结构设计特点，集装箱在船舶上的操作只能是沿竖直方向的操作，如船舶挂靠某个港口进行卸箱操作，而船舶贝位中后续港口的集装箱积压在当前港口想要卸载的集装箱上，则把移走后续港口的集装箱的操作称为倒箱。倒箱的产生增加了岸桥的工作量，配载人员都希望通过合理的方法来减少倒箱的产生。

船舶配积载计划直接影响堆场的取箱顺序故影响着堆场倒箱量，再根据现有堆存状况下制定的合理配积载又会存在船舶倒箱。配载计划制定时，必须考虑当前挂靠港对后续港装卸的影响，当船员无法提前获得想要的信息时，船舶配积载计划无法提前确定；码头工作人员就不会按配积载进行预倒箱，两者之间环环相扣，互相影响。在一些因素影响下时，堆场出口箱区的混堆是不可避免。

以往的一些研究很少将集装箱船的倒箱和堆场倒箱混合考虑，或者简单的以堆场的倒箱作为总的研究对象，这样就忽略掉了当前港口船舶产生倒箱对后续港口船舶倒箱的影响。过去的许多配载都是避免产生船舶混贝位，其实混贝位的存在有时可以提高装载效率，也可以调节集装箱船的安全性能。

因此针对这些不足，通过研究集装箱船、堆场的倒箱问题，提出集装箱船与堆场混合配载优化的方法。首先建立集装箱堆场倒箱模型，并用设计的基于矩阵实数编码的单亲遗传算法求解，通过与其他文献数据比较来验证算法的有效性。然后建立集装箱船与堆场混合配载模型，通过与现今优化方法对比，验证所提出混合优化可行性和有效性。

本文提出的“全航线集装箱船与堆场混合配载优化”，其最终的实现有利于了解在多个港口堆场未做预倒箱情况下，在保证取箱时岸桥贝位间来回移动次数尽可能少的条件下，堆场和船舶配载各自产生的倒箱数量。并在编码设计中，通过扫描法的到堆场取箱顺序，通过产生集装箱装载混贝位，假设这些混贝位能调节船舶安全性能，来确定集装箱在船舶的具体位置。通过综合考虑的倒箱最少来提高岸桥的装卸效率，减少集装箱船在全航线停靠各港口的时间。

本方案的提出，在理论上提出了随机堆箱状态下的取箱优化和船舶配载，为集装箱船舶配载和堆场混合优化的进一步研究提供了参考；在工程

运用方面,相对以往作业模式,考虑将重点确立在某个点,提高国内港口作业效率有一定的积极意义。

1.2 国内外研究现状

1.2.1 装船过程中集装箱配载研究

国外的一些研究者很早就开始研究配载问题在各个领域的应用。研究者希望找到使倒箱量最少的模型以便同时能高效率的完成装卸作业。随着配载问题研究的不断深入,研究者通过添加各种既定约束条件来完善配载的研究,这类问题逐渐被定义为一个多约束多目标组合优化问题^[9]。包括:配载后船舶的稳性、倒箱量、装卸时间及装船前堆场的准备工作等^[1-2]。

Zhao 和 Mi^[3]提出了基于鲁棒方法整数规划模型。模型求解阶段,因考虑的约束条件太多使模型太过复杂,难以根据模型合理应用到实际生产中。

Akio 和 Kaxuya^[4]建立一个整数规划模型。模型中,以船舶稳性和堆场装船倒箱量最少作为约束,并运用遗传算法来优化模型,得到可行解。但该模型并未考虑舱盖的影响,故也很难运用于实际。

Avridl 等^[5]在不考虑船舶稳性下研究最小倒箱量,并归结该问题为 0-1 整数规划问题。

Imai 等^[6]以 GM 最大化为目标建立模型,运用新的算法求取 GM 和倒箱并得到精确值。

Ambrosino 和 Sciomachen^[7]建立了以最小总积载时间为目标建立模型。虽然约束条件过于简单,但也为后面的研究指出了一个新方向。

Dubrovsky 等^[8]提出一种以一个港口的集装箱船舶稳定性为目标函数的模型并通过启发式算法来解决该模型。

而国内在配载方面的研究相对较晚,这或许是因为对集装箱运输的发展不够重视。国内研究者利用数学规划的方法建立配载模型,采用了研究界比较认可的优化方法对问题优化,寻找到最佳配载方案^[9-20]。

张维英^[10-13]2005 年将配载问题细分为箱位布置问题、箱位排箱问题,并分别用基于搜索树算法、禁忌搜索算法求解相应问题,得到不同阶段的最佳答案,为配载问题开创了新思路。2007 年以倒箱最少为目标函数提出一条船挂靠多个港口情况下,贝位中混装集装箱排箱模型,并应用隐式图启发式搜索的方法来优化该模型,得到最优解。2008 年提出一个实用模型。在 05 年的基础上深入研究,提出配载问题进一步分解成贝位选择和贝位中集排箱。2011 年通过禁忌搜索算法来优化新的配载模型,该模型以中心

高度最低、横倾力矩最小为目标函数，为后面的学者考虑船舶性能指明思路。

卫家骏^[14-15]，2009 年提出配积载问题改进算法，取得较好配载效果。在研究时，不考虑 20ft 和 40ft 位置问题，认为所有装载的都是 20ft 或 40ft 集装箱，使问题得到简化。2012 年对智能配载问题进一步研究，建模时采用启发式算法建立堆场优化模型，优化时建立混合蚁群算法来优化。

孙晓雅^[16]，2011 年在假设后续港装载状态信息未知情况下，以倒箱为零的约束条件下，建立了多目标优化数学模型。并在算法方面运用离散粒子群算法进行求解，得到配载结果。

靳志宏，兰辉^[17]，2011 年建立以倒箱最少为优化目标的模型。该模型假设知道堆场信息、船配载图，创新点在于考虑海关是否放关。在优化模型时用启发式算法优化，得到最佳染色体。

曲目，计明军^[18]，2011 年提出倒箱量最少为目标函数的模型，选用算法时通过遗传算法进行求解。得到了在保证船舶安全性能的条件下，配载优化方法，但是，其只考虑了一船装，多港卸的情况。

王君红，计明军^[19]，2013 年用遗传算法优化建立的以倒箱量等多个目标为目标函数模型，设计了将船舶各贝位以舱盖划分成 6 个区间考虑其倒箱和安全性能的方法对集装箱装船顺序进行优化的遗传算法，最后检验可行性和有效性。

祝慧灵，计明军^[20]，2014 年，在集装箱船稳性、强度、载荷约束下，建立全航线倒箱最少、吃水差最小的模型，并利用启发式算法和遗传算法优化。

在配载优化问题研究方面，很多因素制约着其研究。以上的参考文献或多或少都有考虑不周的方面，如对于集装箱堆场堆存情况的考虑不足、对某些因数的未考虑以及假设后续港口装载信息已知等。制定配积载计划时应该充分考虑到堆场混堆的影响。

1.2.2 关于堆场堆存优化研究

目前堆场倒箱方面的研究，已经有许多，但是研究者都避免在出口箱区混堆贝位的产生。当同一贝位内混堆各港口卸载的集装箱时，倒箱是待解决关键问题。如何改进堆场堆存集装箱的模式、如何更合理有效分配岸桥调度集装箱、如何确定规划的靠泊船舶的集装箱装船顺序、如何积极寻找调度新方法等成为港口工作人员及研究者一直关心的问题。目前堆场堆存优化方面主要是岸桥调度、倒箱等。

陈庆伟等^[21]简单的考虑出口箱区混堆模型，运用启发式算法求解该模

型得到想要的结果。该研究解决了考虑堆场堆位中非同类箱子混堆情况下的配载问题，但因考虑因素太少，不便于实际操作。

卫家骏^[22]运用启发式算法来优化在倒箱为零等多个约束条件下建立的集装箱堆存位置的模型。该研究还结合了人工经验。

周鹏飞等^[23]将箱位选择分成堆场分配和具体箱位选择，建立两个不同的倒箱最少的目标函数，并用禁忌搜索求解问题。该问题的提出建立的假设条件为不知到船舶到港时间。

刘庆^[24]建立码头船舶配载和堆场规划采用同贝位边装边卸方法，但此方法只是定性的说明，并没有运用模型求解。

王秀臣^[25]建立了协调优化模型来求解随机入港下的船舶配载问题。在算法设计方面，综合前人的研究思路采用模拟退火算法来解决问题。

朱明华^[46-48]根据集装箱到港的随机性，综合考虑堆场的堆放位置和船舶贝位之间的对应关系，并建立了堆场模型和船舶的配载方案，在算法的优化上为配载的研究提供了理论依据。在装船前定义同类箱组，为船舶的配载提供了便利，并用启发式算法和蚁群算法结合的方法求解了堆场堆放的集装箱数量和船舶贝位数量之间对应的关系。

集装箱船舶随机入港情况下，出口集装箱具体信息无法提获取，造成船员无法将船舶配积载计划确定，码头因此来不及对出口箱区进行预倒箱，造成堆场出口箱区混堆情况。针对这些问题，本文提出随机产生出口箱区混贝位的堆箱状况，优化取箱顺序的同时减少取箱时岸桥贝位间来回的大距离移动次数，来减少单船进出港时间，并调节集装箱船的配载来保证船舶尽可能少的产生倒箱的思想。

1.2.3 关于遗传算法的研究

以上研究者研究配载问题时，提出了在各种条件作为约束条件下的配载优化模型，并通过不同优化方法来实现研究。本文要采用遗传算法（GA）来优化模型，并在编码设计时有所创新。

遗传算法是通过染色体的进化来模拟生物进化的一种人工智能技术。遗传算法的运作流程可以描述为：从任一初始化群体出发，执行各种遗传操作如单条染色体的变异、两条染色体的交叉等，并筛选得到最优适应度函数，选取最优的染色体的过程。目前研究表明用单亲遗传算法（PGA）求解序号排列最优化问题无论是在其编码的便捷性，计算效率方面，还是其收敛性情况，都比传统遗传算法（TGA）有优势^[26-35]。使用遗传算法首先要通过基因编码想要解决的问题。为了解决编码问题，可以选择不同的编码方式，如二进制编码，实数编码等，研究者往往会根据具体情况选用

合适的方法。

李茂军，童调生^[29]指出不同的编码方式串空间的大小有很大差别，选择串空间小的编码方式有利于提高搜索效率。将同一组合优化问题按不同的同序基因数来编码，同时还介绍了单亲遗传另外集中编码方式，为编码多元化提供前进动力。

陈慧琴等^[30]根据 N 皇后的例子具备的特性，选用整数编码来解决该问题，该研究能很好体现单亲遗传相对传统遗传算法的优越性，为以后配载问题选用单亲遗传算法作铺垫。

1.3 本文研究内容

本文通过研究国内集装箱船配载现有的模式，进一步完善堆场与船混合配载方法，以宁波港吉公司为试点单位，通过获取实际信息和数据处理后应用到数值模拟中，来完成工程验证。主要包括：

（1）集装箱堆场出口箱区混堆的优化方法

在全航线下，根据堆场的现存堆存模式，考虑到取箱顺序对船舶配载的影响，以倒箱量最小、岸桥贝位间来往复移动次数最小为优化目标，建立出口箱区取箱模型，并用扫描法编程优化，得到最佳堆场取箱顺序输出于单船配载优化。

（2）单船全航线配载与堆场混合配载优化方法

根据集装箱船舶的结构特点，以集装箱舱盖位置将集装箱船舶舱位进行分区，共分 6 个区。贝位内装载同一目的港和混合目的港的集装箱。合理排箱，保证船舶安全性能及舱盖下不倒箱等前提下减少倒箱量，形成混贝位的装箱情况。

研究在不同约束条件下，单船在多个港口挂靠，以倒箱量最少为目标，建立堆场与船混合配载模型。并考虑到船舶后续港口尽量少倒箱、堆场取箱造成场桥贝位间往复移动次数尽可能少，通过（1）中得到的堆场堆箱来优化算法求解，运用遗传算法求解并与单船优化结果对比，其结果表明混合模型的有效性。

（3）优化方法

编码方面选用单亲遗传算法。为了简化编码难度，选用较简单的实数编码的。编码内容包括：出口箱区混堆矩阵、扫描法取箱方法、船舶配载、堆场和船舶倒箱记录、场桥贝位间往复移动次数等。最后通过 Matlab 软件编程完成实现理论问题分析，并采集实际数据验证理论。

1.4 论文通篇结构

第一章 绪论。介绍论文研究背景及意义。主要包括装船过程中集装箱配载研究现状分析、堆场堆存优化研究现状分析、遗传算法的研究现状分析。

第二章 集装箱堆场及集装箱船概述。主要内容为介绍堆场定义；堆场主要装卸设备；集装箱在堆场的堆存；堆场倒箱以及配积载定义、原则；坐标表示；船舶倒箱。

第三章 集装箱船与堆场混合配载方法模型的建立。从问题描述到数学建模以及其中的各种假设条件。

第四章 本章在第三章研究的基础上，通过遗传算法的算法优化计算，验证混合配载优化方法的有效性。

第五章 工程应用。以宁波北仑港吉公司为试点，通过获取实际信息和数据处理后应用到数值模拟中，来完成工程验证。

第六章 总结与展望。

2 集装箱配载涉及内容相关概述

集装箱配载涉及很多内容，其中集装箱码头地位尤为重要。集装箱码头(Container Terminal)功能强大，可作为船舶停靠地，还可以作为集装箱中转、集装箱堆放地。如图 2.1 所示，码头主要包括泊位(如图 2.1 中的 1)、码头前沿(如图 2.1 中的 2)、集装箱堆场(如图 2.1 中的 3)、控制室(如图 2.1 中的 5)、检查口(如图 2.1 中的 4)等多个部分^[37]。本章节将主要介绍集装箱码头中的堆场、船及与两者相关的其它内容。

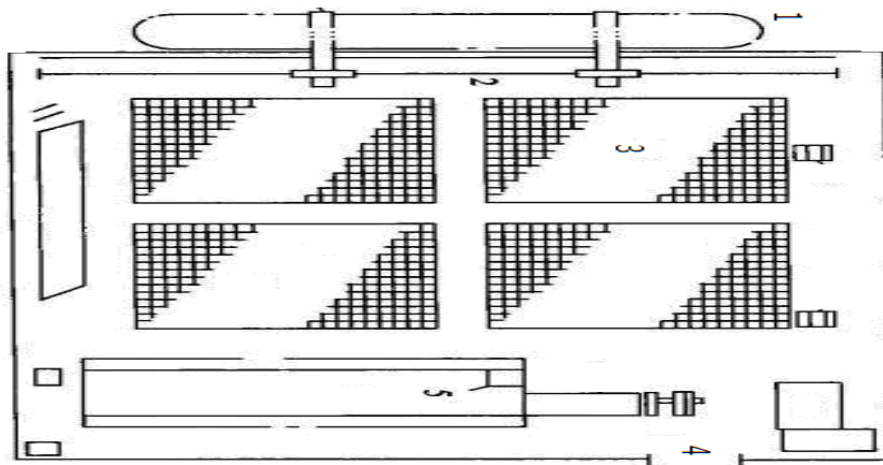


图 2.1 集装箱码头

Figure 2.1 Container terminal

2.1 集装箱堆场相关概述

2.1.1 堆场定义

堆场是指办理集装箱进行中转、堆放、缓存等一系列操作的交场所。堆场按照功能和布局可分为两大块：前方堆场和后方堆场^[38]。前方堆场指靠近码头前沿的场地。集装箱船到港前，有计划的将出口箱堆放到前方堆场有利于出口箱的装箱，同时将进口箱堆放在前方堆场有利于船舶进口集装箱更快的卸载，由此可见前方堆场加速船舶装卸作业速度；后方堆场是集装堆存保管的场所。当船舶卸载集装箱后，集卡可从前方堆场提取需要提交、中转运输方式的集装箱堆放到后方堆场，待后续人员来提取。

2.1.2 堆场装卸设备

港口堆场是个复杂地，堆场内运行的机械装卸设备有很多，经过几代的发展机械化早已实现。堆场运送集装箱的设备包括：分为水平运输设备和场地装卸设备^[38]。

其中水平运输设备统称为拖车，主要完成集装箱的水平运输工作。拖

车是集装箱转移的重要设备之一，如作为后方堆场到出口箱区、进口箱区到后方堆场的运输工具。

场地装卸设备主要用到场桥，又称为龙门吊。场桥有轨道式场桥和轮胎式场桥，是用来运输堆场内集装箱、抓取堆场内集装箱的机械设备。各大港口出于作业环境的影响考虑一般会选择轨道式场桥，另外港口还会用到正面吊和空箱叉等机械设备，这些设备相对场桥更灵活，设备低廉，但工作效率低、经常维修等问题让其难以重用。

当运输某个集装箱时，水平设备和场地设备一般都是结合运用的，即用场桥（场地设备）将堆场内的将要出口的集装箱装载到集卡（水平设备）车上，通过集卡运输来实现码头前沿与后方堆场的连接。

码头前沿的装卸设备统称为岸桥，岸桥指岸边用的桥式设备。目前各港口码头前沿常用的设备为岸壁式装卸桥，来将集装箱从船舶装卸到码头、将码头集装箱装船。此外岸桥还包括 3 种：双 20 英尺装卸桥、双 40 英尺装卸桥、三 40 英尺装卸桥，但相对选用指数较少。

2.1.3 集装箱堆场堆存状态

集装箱堆在堆场中的某处，工作人员需对其进行管理和其他操作，必须知道集装箱的具体位置，根据规定以箱位定义为堆场中的最小单位^[36]来表示集装箱的具体点如图 2.6 所示。其中箱位还可以细分为以下概念：定位箱区、街号、区号、贝位。这样集装箱在堆场的具体点就可以用一个 8 位数的数值串表示。

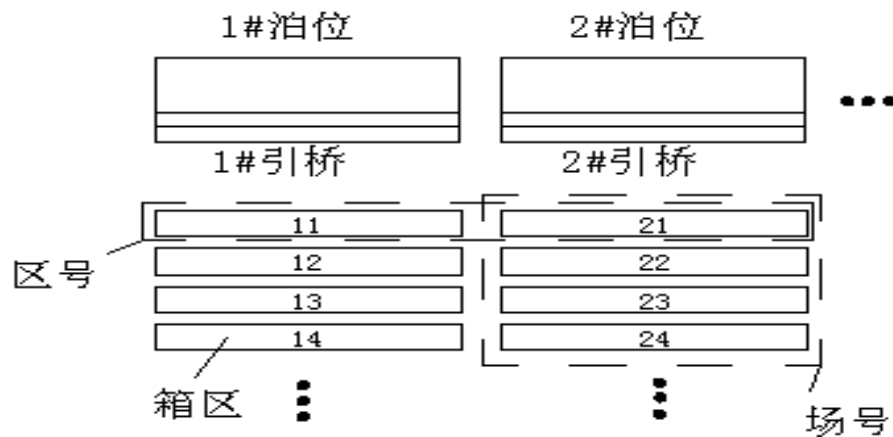


图 2.6 堆场分布

Figuer2.6 Container yard distribution

定位箱区：用 8 位数值串的前两位表示集装箱在堆场内具体的某个箱区。其中 8 位数数值串的第一位表示箱区中的场号，而第二位数值表示箱

区中的区号。从图可以看出堆场被分割成多个箱区。

场号：用集装箱在箱区纵向位置表示场号，场号一一对应码头泊位号。

区号：对应着集装箱在横向的位置。其中以靠近泊位一侧到陆侧后方依次按顺序排列。

贝位：用 8 位数值串中的三四位置表示箱区中的一列内所有集装箱，即贝位，如图中的 11、12。贝位还可以细分到栈和层即 8 为数值串的五六合七八两个位置。

定义好了以上概念，就可以数值串来表示堆场中集装箱的具体点，工作人员进行操作将会更方便。

2.1.4 堆场倒箱

上文提到贝位细分可以分到栈，场桥对堆场各栈进行提箱操作时，可以分为水平取箱操作、垂直取箱操作及水平竖直混合取箱操作。若只考虑垂直取箱问题，刚好堆场贝位出现非同类箱区混堆，此时极有可能产生倒箱操作。如图 2.7，现需提取 9 号箱，因 9 号箱不在该栈最上层，移走到 17、13 号箱去 4 栈，此时产生倒箱。随后需要取其他集装箱，又需要把转移的 17、13 号箱再进行一次转移的操作叫做二次倒箱操作。二次倒箱的操作，极大的增加了场桥工作时间，操作人员须避免。

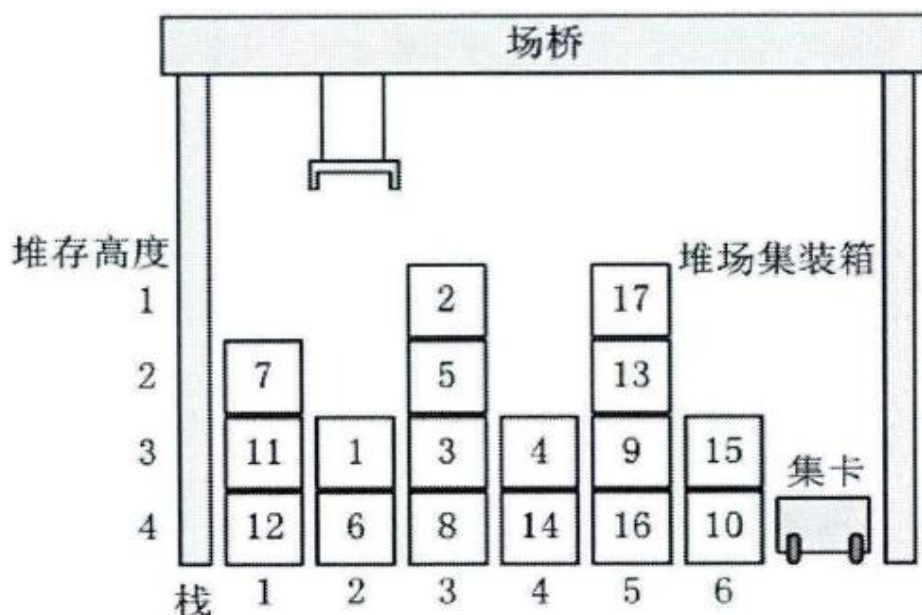


图 2.7 堆场提箱图

Figuer2.7 Loading container for container yard

由于工作人员会将相同目的港的集装箱堆放到同一贝位中，这样就不会产生倒箱问题，但会降低堆场容量。但集装箱船集港时，进入堆场顺序

的随机以及各种原因，总会产生混贝位，由此产生的倒箱问题，不容忽视。

造成堆场倒箱的主要原因有：

（1）不合理转移阻塞箱。当堆场的箱位紧张时，工作人员不得不将集装箱混堆来提高堆场利用率从而产生的倒箱，以及在取箱时的不合理的转移造成二次倒箱等；

（2）不合理的船舶配积载。忽视装箱在堆场的堆存状况、出口箱区未预倒箱等因素都会造成配载计划的不合理，从而增加堆场倒箱的产生。

（3）入港集装箱的不合理堆放。在集装箱集港过程中，由于工作量太大，来不及进行系列操作，集装箱到达堆场的顺序会变得更随机，导致了不同重量级集装箱混堆。

（4）不合理的提箱顺序。操作人员在没有得到具体的配载计划时往往根据经验确定提箱顺序，很难制定理想取箱顺序，也会导致倒箱。

2.2 集装箱船相关概述

1957 年集装箱船舶出现，随着发展成为一种以装载集装箱为目的的多设计为艙机型的特定船舶，设计时满足装载 20ft、40ft 的集装箱。经过迅猛的发展，集装箱船出现了多元化的发展，按装运集装箱程度分：部分、可变换、全集装箱船；按照装卸方式可以分^[39]为：滚装式、浮装式、吊装式集装箱船；按照用途分为：破冰集装箱船、一般集装箱船。

本文研究的内容涉及的船舶为全集装箱船。全集装箱船是一种为满足只运输集装箱而设计成 U 型大开口的船舶。由于设计时为了能装载更多集装箱其本身没有装卸设备，而是利用码头上的岸边装卸设备（岸桥）进行垂直方向的装卸操作。

2.2.1 集装箱船配载的概念

集装箱码头每天会考虑不同的原因，制定船舶的各种配载问题。配载图能保证船舶的安全性能前提下，将集装箱尽可能多的配载到船上，并满足全航线船舶挂靠港口倒箱尽可能少，提高装卸效率。所以需要在船舶还未装卸前就制定具体箱位装船位置的配积载图，保证满足全航线的装载要求。船舶配载可以分为以下几个步骤，如表 2.1 所示。

表 2.1 集装箱船配载流程

Table 2.1 Container ships loading process

收齐核对单证、收集各种有用资料	
编制配载船图	
稳性和吃水差的计算	
审核配载船图	
大副签字的配载图	

2.2.2 集装箱船配积载原则

工作人员进行配载工作时，一般都会遵循配载原则，配积载的原则^[49]主要包括：

（1）保证船舶良好的稳性。稳性是指船舶受到外界力量的作用，发生船体倾斜后外界作用在船体的作用力突然消失，船舶自行回复到原来位置的能力^[49]，回复得越快说明稳性越好，但船舶设计时不宜过分考虑稳性，还要考虑横摇周期。研究规定用稳性高度（GM 值）来衡量船舶稳性好坏。集装箱装载集装箱后，受风面积大增，满载时重心高度较高，GM 值过小，此时除了用压载水来调节重心高度外，还要遵循“重不压轻，后不压前”的原则，以提高初稳性高度；

（2）保持船舶适当的吃水差。吃水差是指船舶艏艉吃水差值^[50]。不适当的吃水差会造成使船舶航行性能的耗损，使主机功率得不到充分发挥，造成经济损失和安全性能丢失。不同的港口、码头配对船舶的吃水有不同额要求，集装箱船应尽量减少压载水调节吃水差的方法以便船舶挂靠港口更方便；

（3）满足船体强度要求。一是工作人员在配载集装箱工作时，不仅要让总的集装箱量满足船体总体强度，更应让贝位甚至细分到栈在配载的集装箱时不超过船舶允许的设计规定的堆积负荷满足局部强度；二是船舶要保持适当纵向强度，防止船体中拱的发生。若艏部满载会造成船体中拱，船舶若承受不住中拱产生的剪力，船体就会断裂。配载时，可适当将后续港口的重箱多配置在船中，用来抵抗中拱的产生；

(4) 避免中途倒箱。全航线集装箱船挂靠多个港口进行装卸集装箱，此时配载工作得综合考虑整条航线的倒箱量，虽然部分港口的倒箱量会相对增加，但总体倒箱可以保证最少。若只考虑当前港口倒箱量而不考虑整条航线的倒箱，船舶在后续港口产生的倒箱会让岸桥来回移动的次数更多；

(5) 避免同目的港箱子过分集中。港口装卸桥的结构特点是结构大、重型化，考虑到操作安全不能让 2 台装卸桥并在一起作业，一般须相隔 2 个贝位以上安全距离。同目的港集装箱配载时过分集中，不仅只能由 1 台岸桥工作闲置其他岸桥，大大降低装卸效率，还会在集装箱全部卸载完成后，影响船舶安全性能。遇到某一卸港的箱量较大，超过一个船容量而必须分舱时，要避免形成”重点舱”，应将集装箱配置在至少相隔 2 个贝位的位置，这样不仅满足多台装卸桥同时作业，还可以保证船舶稳性；

(6) 满足特种箱配载要求^[49]。

为了得到科学的配载方法，需掌握配载所需资料，包括：

(1) 堆场集装箱资料，如表 2.2 所示；

表 2.2 堆场集装箱资料

Table2.2 Container yard information

集装箱装箱单	←	堆场 集装 箱资 料	→	危险货物清单
装货单	←		→	预配船图
特种箱的清单	←		→	集装箱在堆场中的箱位

(2) 集装箱船舶资料，如表 2.3 所示；

表 2.3 集装箱船舶资料

Tab 2.3 Container ship information

船舶箱位布局和容量	←	集装 箱配 载所 需资 料	→	冷藏、危险箱的装载规则
船舶参数	←		→	稳性、吃水差计算书
堆积负荷强度	←		→	空船重量和常数

(3) 船图的表示方法。

船图表示方法即用来表示集装箱在船上的具体位置，下文会在需要具介绍的位置展开论述。

2.2.3 集装箱船坐标系

集装箱在船舶上的表示方法可按 ISO 委员会规定了一个标准来表示集装箱船上每一个集装箱的具体位置^[9,52]。船舶坐标系定义如图 2.9 所示。其中船舶坐标轴相对应一一对应到船的贝、列、层，贝位 \in {相同 x 坐标的箱位}；列 \in {相同 y 坐标箱位}；层 \in {相同 z 坐标的箱位}。通过一一对应关系后可以用一个 6 位数值串来表示船舶具体的每一个箱位。

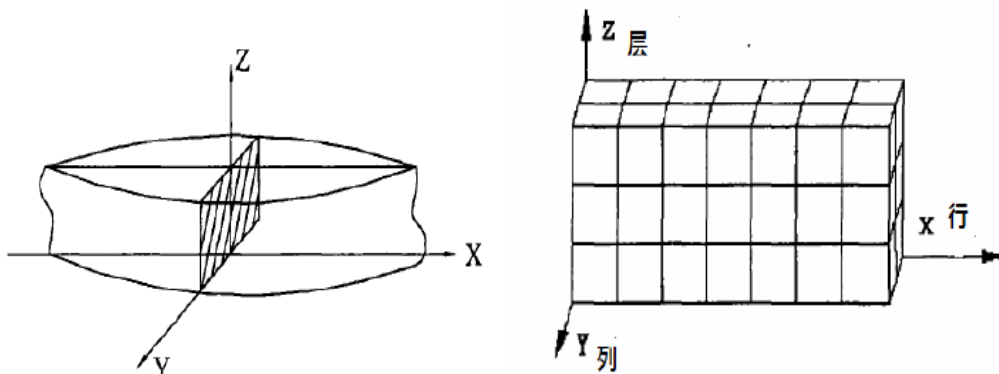


图 2.9 船舶坐标系

Figuer2.9 The ship coordinate system

贝号 (Bay number)。用 6 位数的前两位数字表示船舶纵向截面从船首向船尾方向排列的集合^[9]，即贝号。从船首至船尾依次编号为 01、02、03……。行位分 20ft 和 40ft 两种，如图 2.10 所示。奇数按规定表示 20ft 集装箱贝，偶数按规定表示 40ft 集装箱贝。从船首至船尾依次编号为 01、02、03……。

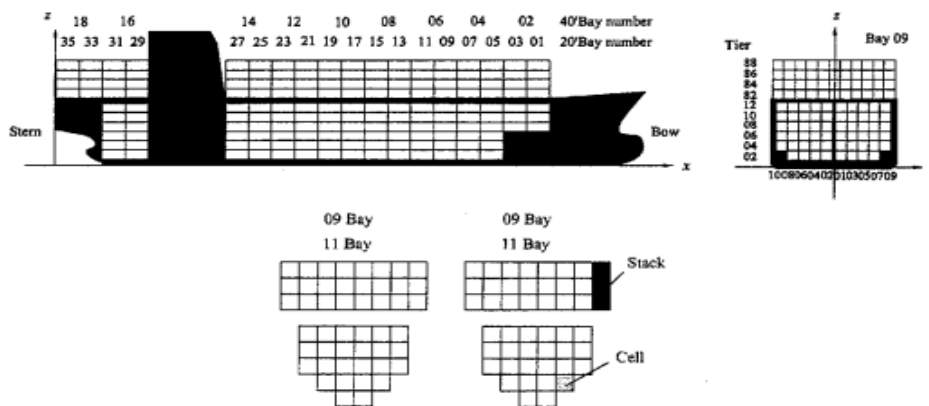


图 2.10 船舶结构和贝位布局

Figuer2.10 Structure and view of bay

列号 (Row number)。6 位数的中间两位表示船舶横剖截面方向排列的箱位的集合^[9]，即列号。列号可以有两种表示方法：一种是以船舶纵向中轴为基准，中间列编号为 00，向右舷方向用 01、03、05……等奇数依次编号，向左舷方向则依次用 02、04、06……等偶数编号；另一种船舶右舷向左舷依次用 01、02、03……编号表示。本文采用第一种方法表示。

层号 (Tier number)。6 位数的后两位表示横剖截面上箱位的上下排列集合即层号。可分为两种表示方法^[9]：一种是从船底最下一层向甲板最顶层依次编号，即 01、02、03…；另一种表示方法是舱内的层编号前面加“H”，甲板上的层编号前面加“D”，用来区别舱内和舱外。舱内编号从下往上依次 H1、H2、H3……，而舱外则是依次为 D1、D2、D3……。本文选用第二种表示方法。

建立好船舶坐标系，可以用一个 6 位的数值串表示船体的每一个箱位，如 050303 表示集装箱的位置在第 5 个 bay，右舷第 2 列第 3 层的箱位。

2.2.4 集装箱在船舶上的布局

本文为了介绍集装箱在船舶的布局，将船体结构继续细分引入舱位、堆垛的概念^[36]，如图 2.11 所示。

舱位(Hatch)：以仓储盖为标准，仓储盖区域内堆放集装箱的位置称为箱位。

堆垛(Stack)：舱位中堆放集装箱的每一列都称为一个堆垛。根据船舶分区概念下可分为舱位盖上、下堆垛。

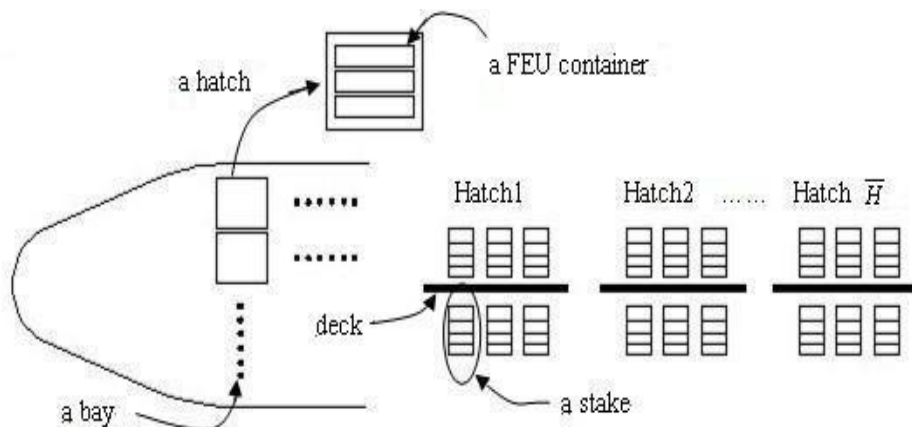


图 2.11 集装箱在贝位中位置

Figuer2.11 The layout of the container ship's bay

2.2.5 集装箱在船上倒箱

出于集装箱船舶的结构特点，集装箱的装卸只能是垂直方向的操作，因此，如同堆场堆箱产生倒箱一样，集装箱配载于船上也会产生倒箱。假设全航线船依次挂靠港口 a、b、c，当非同类箱装载到同一个贝位时，因考虑到船舶的安全性能等，重箱装载到较低的层位，轻箱装载到较高的层位。此时在 a 港口装载 b、c，若 b 是重箱 c 是轻箱，会造成 c 堆在 b 上，船舶到达 b 港口要卸载 b，此时会产生倒箱。

集装箱船舶在装卸过程中产生倒箱的原因主要有^[25]三个：一是全航线船舶大量装箱，配载时要综合考虑很多因素，增大了减少倒箱问题的难度；二是出于调节船舶安全航行在装船过程中进行调箱造成的倒箱；三是发现配载情况不好后在某挂靠港口做适当调整产生的倒箱。

2.3 本章小结

本章介绍了集装箱配载的相关概念，介绍初级堆场定义；堆场主要装卸设备；集装箱在堆场的堆存；堆场倒箱的产生以及集装箱船配积载时的定义、配载原则；船舶坐标表示方法；贝位内结构布局的定义；船舶倒箱的原理及原因。为下问构数学模型提供一定的理论知识基础。

3 全航线集装箱船与堆场混合配载模型建立

3.1 问题的提出

工作人员在实际配载装船操作时只是将配载图作为参考，更多的以自身积累的经验来综合考虑更多因素，确定提箱顺序进行装船。这样做或多或少会增加装船时间，让船舶在港时间增加，而且就算满足船舶稳性、吃水差等要求但会增加船舶倒箱并影响下几个港口的配载。因此需要对其优化。

堆场取箱顺序与船舶配积载是相互影响，船舶的配积载计划决定了下一港口出口箱区的队箱顺序，而下一港口堆场堆放又决定了下一个船舶配载的计划，在未知船舶预配的情况下为了减少取箱过程中堆场倒箱和场桥移动距离、提高装船作业效率，在已知各种信息时，港口就会针对的进行预倒箱，针对未知下一港口的预配问题则需要考虑船舶和堆场倒箱之间的平衡。

本文要研究的就是利用遗传算法来优化堆场的取箱顺序，在港口未预倒箱情况下，随机产生各港口的混堆情况，通过选取合适染色体使得堆场出口箱区倒箱少的同时配载到船上的集装箱的倒箱量少，以其整体的倒箱作为衡量标准。其中取箱时，尽可能减少场桥的贝位间来回移动次数产生。从而提高装船速度和码头工作效率。本文根据现实堆场的堆存情况，考虑出口箱区混堆的倒箱问题，以集装箱船舶的容量及不超过集装箱装卸容量等为约束条件，在单船多港口全航线下，以集装箱船与码头堆场的总倒箱量最小和取箱时场桥贝位间往复移动次数最小为优化目标函数，建立集优化模型。

3.2 基本内容

3.2.1 运输矩阵及同类箱组定义

本文模型建立中，需要得到各个目的港口装载的集装箱数量，定义运输矩阵 $T^{[10-13]}$ ：

$$T = \begin{bmatrix} T_{12} & T_{13} & \dots & T_{1P} \\ T_{22} & T_{23} & \dots & T_{2P} \\ \dots & \dots & \dots & \dots \\ T_{P2} & T_{P3} & \dots & T_{P1P} \end{bmatrix} = \begin{bmatrix} T_{12} & T_{13} & \dots & T_{1P} \\ 0 & T_{23} & \dots & T_{2P} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & T_{P1P} \end{bmatrix}$$

T 表示在港口 $p(p=1, 2, \dots, P-1)$ 进行装载集装箱, 在港口 $j(j=2, 3, \dots, P)$ 进行卸载集装箱。通过变形整理得到新的运输矩阵, 本文的后期编码, 需已知运输矩阵。

为了统一各港口集装箱的重量、货物、目的港等问题, 定义同类箱组^[12]: 装载的是相同类型货物、相同尺寸、货源港及目的港相同的集装箱的集合。确立了同类箱组的问题, 各港口的集装箱可以有机结合, 进行某些计算可忽视集装箱重量。

3.2.2 堆场出口箱区矩阵级场桥多余移动次数定义

为方便优化算法的编码, 根据运输矩阵取各各港口的集装箱, 现定义出口箱区堆箱矩阵为 g :

$$g = \begin{bmatrix} d & a & c & \dots & \dots c & b & b \\ c & a & d & \dots & \dots c & c & b \\ d & c & d & \dots & \dots b & a & d \\ d & b & b & \dots & \dots a & a & d \\ b & b & b & \dots & \dots a & c & 0 \\ c & a & a & \dots & \dots a & a & 0 \end{bmatrix}_{6 \times N}$$

g 表示各各港口出口箱区集装箱堆放情况。如 $g1$ 表示第一个港口出口箱区堆存状态; $g2$ 表示第二个港口出口箱区混堆的堆存状态, 依次类推。其中 a, b, c, d 分别代表目的港为 2, 3, 4, 5 港口的集装箱 N 表示矩阵的长度, N 的数值计算定义: 运输矩阵在各港口的运输总量/6; g 中数字表示各目的港口的集装箱, 并不是表示数量。如例子 $g1$ 表示港口 1 混堆情况。矩阵中的数值 4 表示 4 号港口的某个集装箱, 0 表示该处不存在集装箱, 只是为了方便编码成矩阵的添加数; 6×20 ($6 \times N$) 表示在港口 1 一共装载的集装箱数量。

$$g1 = \begin{bmatrix} 4 & 2 & 3 & \dots & \dots & 3 & 2 & 2 \\ 3 & 2 & 4 & \dots & \dots & 3 & 3 & 2 \\ 4 & 3 & 4 & \dots & \dots & 2 & 4 & 5 \\ 4 & 4 & 2 & \dots & \dots & 3 & 4 & 5 \\ 2 & 4 & 2 & \dots & \dots & 2 & 5 & 0 \\ 5 & 4 & 2 & \dots & \dots & 2 & 3 & 0 \end{bmatrix}_{6 \times 20}$$

为了记录场桥的贝位间往复移动次数 K ，现在假设出口箱区每个贝位为 6×6 的矩阵（若堆得箱数不够，用 0 表示），这样可将各港口堆箱看成集装箱堆在不同贝位中，如图 3.1 所示。

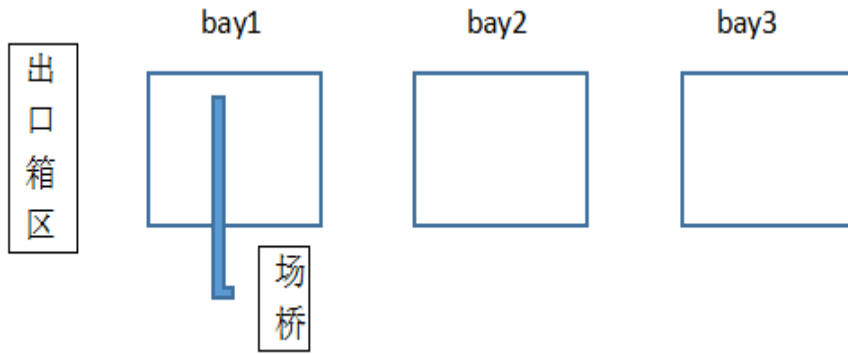


图 3.1 出口箱区堆箱

Figure 3.1 Export containers in container yard

扫描取箱时，若合理取箱，场桥只会在每个 bay 的安全距离上行进一次；若不合理的取箱，场桥会在贝位间往复移动，严重影响效率。

如随机选取的 $g3$ 港口的出口箱区，见图 3.2。可以看成港口 3 的出口箱堆在该港出口区的 bay1, bay2。取箱时，场桥若来回移动于贝位 1 和贝位 2 之间，会严重影响效率，而场桥在某个贝位内的来回移动并不会太大影响。故本次需要记录的是场桥在贝位间来回移动次数，用 K 表示。

$$g3 = \left[\begin{array}{cccccc|cccccc} 4 & 4 & 4 & 4 & 5 & 4 & 4 & 5 & 4 & 4 & 4 & 4 \\ 5 & 5 & 4 & 5 & 4 & 4 & 5 & 4 & 5 & 4 & 4 & 5 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\ 5 & 5 & 5 & 4 & 5 & 5 & 4 & 4 & 4 & 4 & 5 & 0 \\ 4 & 4 & 5 & 4 & 5 & 4 & 5 & 4 & 5 & 5 & 5 & 0 \\ 4 & 4 & 4 & 4 & 4 & 5 & 4 & 4 & 5 & 4 & 5 & 0 \end{array} \right]$$

图 3.2 港口的堆箱示例

Figuer3.2 The example of port's container

3.2.3 权重 λ 的确定

为确定场桥贝位间往复移动次数与倒箱数之间的权重 λ ，选用两种不同方案取箱 A-B-C 得到相同的效果，如图 3.3。其中 S 为跨 bay 位的安全距离， L 为场桥取箱移动距离， m 为两栈的间距。

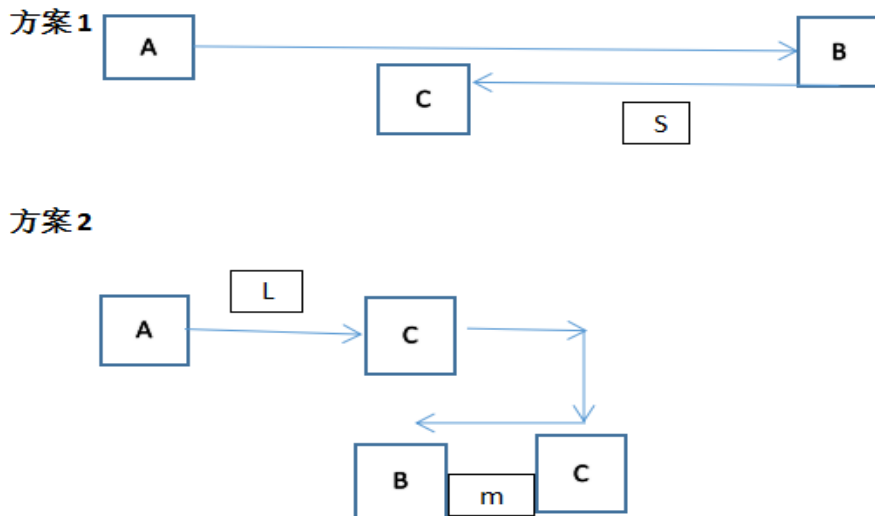


图 3.3 不同取箱方案

Figure 3.3 Different loading scheme

方案 1 场桥跨 bay 完成 A-B-C 的取箱操作；若选择不跨 bay，方案 2 则操作路线是 A-C-B-C（因 C 箱堆压 B 箱，故要先移走 C 箱）。确定两者权重：1 方案操纵路线时间/2 方案时间。

方案 1 路径距离 $a1=2S+L$ ，方案 2 路径距离 $a2=L+2m$ ，两种方案同时除以场桥的平均移动速度（考虑到场桥在 bay 内移动速度不等于跨 bay 移动速度）后，做商比较即可得到结果。

$$\begin{aligned}
 l &= (a1/v) / (a2/v) \\
 &= (2s+L)/(L+2m) \\
 &= 4.6
 \end{aligned}$$

本文考虑倒箱为整数，目标函数计算后得到的结果应为整数，故 l 取5。

3.2.4 船舶箱位的表示方法

集装箱船舶设计为U型，其横剖面形状是不规则的，用 (i,r,t) 表示具体某一贝的第 r 列第 t 层的箱位来具体表示船舶贝位中的箱位。为形象表示其截面图，假设集装箱船舶的贝位都由R列T层的矩形组成，每一个小矩形代表一个集装箱箱位。用0表示该箱位存在用，若不存在用-1表示。表3.1为贝位内箱位图示例。

表 3.1 贝位内箱位图
Table3.1 Cells denotation in the bay of ship

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0	-1
-1	-1	0	0	0	0	0	-1	-1
-1	-1	-1	0	0	0	-1	-1	-1

3.2.5 船舶分区表示方法

集装箱船舶分成甲板 and 船舱两个部分，一般每个贝位对应3个舱盖，将贝位分成6个区如图3.4所示。本文用 $n(i,s,h)$ 表示集装箱箱位集合。其中， i 即下文参数设计中船舶贝位号集合； s 为区间号集合； h 为分区后上

下两块，即 $h=1$ 表示舱盖以上部分， $h=0$ 表示舱盖以下部分。船舶虽然分区，但根据装载原则，舱盖下部分一般避免产生倒箱，故倒箱主要产生在舱盖以上部分。下文模型求解部分以此根据考虑并将船舶 bay 看作 $4*6$ 的矩阵。

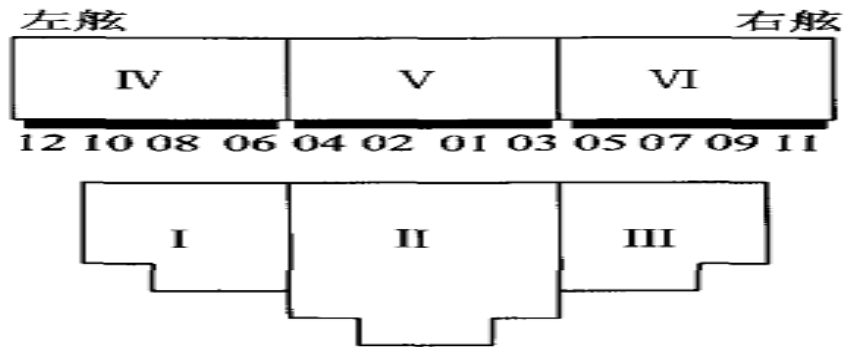


图 3.4 船舶的分区图

Figuer3.4 Maps of the ship bay

3.3 混合配载模型建立

3.3.1 模型假设

集装箱船与堆场混合配载优化问题是一个 NP 问题^[9-10]，为了简化模型，提高方案的运行速度，现作如下假设：

- (1) 集装箱船在起始港装载集装箱后，中途挂靠港口卸载该港口的目的箱并装载后续港口集装箱，直到第 n 个港口全部卸完；
- (2) 假设集装箱的装、卸作业是相互独立的，即后续港口卸箱操作完成后才进行装箱操作；
- (3) 假定配载计划时不需要考虑冷藏箱和特殊箱；
- (4) 假设每次装载出口箱时场桥参与数量视集装箱数量而定，一般情况为单台操作；
- (5) 假设船舶一次航行停靠 n 个港口，到每个港口的取箱信息已知，但堆场未做预倒箱；
- (6) 假设船舶每个贝位的箱位为 $4*6$ 的矩阵；
- (7) 假设整个配载过程中的混贝位产生满足调节船舶安全性能；
- (8) 假设船舶待装载的集装箱都为同尺寸的集装箱，即都是 40ft，或 20ft 集装箱；

3.3.2 参数和变量设计

在本文的数学优化模型中，以总的倒箱最少、取箱时岸桥贝位间来回

移动次数最少为目标函数建立模型。整个建模过程中定义的参数和变量说明如下：

i 表示集装箱堆存在堆场贝位集合， $i=\{1,2,3,\dots,I\}$ ；

r 表示集装箱在堆场列集合， $r=\{1,2,3,\dots,R\}$ ；

t 表示集装箱在某贝位的层集合， $t=\{1,2,3,\dots,T\}$ ；

$P、J$ 表示停靠的港口集合， $p、j=\{1,2,3,\dots,P\}$ ；

i' 表示集装箱配载在船舶的贝位集合， $i'=\{1,2,3,\dots,I'\}$ ；

r' 表示集装箱在船舶贝位的列集合， $r'=\{1,2,3,\dots,R'\}$ ；

t' 表示集装箱在船舶贝位的层集合， $t'=\{1,2,3,\dots,T'\}$ ；

$a_p(i,r)$ 表示在 p 港口，堆场 i 贝位中 r 列的集装箱量；

$x_{pr}=1$ 表示在在 p 港 r 列有倒箱，否则为 0；

j_i^L 表示出口想区贝位 i 中待装集装箱的数量；

j_i^D 表示进口箱区贝位 i 中待卸集装箱的数量；

j_i^s 表示贝位 i 中参与装卸操作的集装箱；

M_p 表示计划装载集装箱量；

N_p 表示计划装载集装箱量；

$n(i,s,h)$ 表示船舶集装箱箱位；

a_{pis} 表示在港口 p ，船舶 i' 贝位 s 区间上的箱数量， $a_{pis} \in (QA)$ ；

i' Bay 有 R 列、 T 层，箱位的表示 $ce(l,r,t)=-1$ 表示箱位不存在，否

则 $ce(l,r,t)=0$ ，箱位最大容箱量为 $A=RT\sum_{r=1}^R\sum_{t=1}^Tce(l,r,t)$ ；

D_{pj} 表示挂靠港 P 时装载港 J 的集装箱数量；

F_{pi} 表示挂靠港口 P ，贝位 i 中空箱数量；

$H_p(i,s)=1$ 表示在 i 贝位，第 s 区间产生倒箱，否则为 0；

K 表示出口箱区取箱时，岸桥贝位间来回移动次数。

3.3.3 优化目标及约束

基于以上假设条件，本文将建立全航线集装箱船与堆场混合配载模型。主要介绍的是目标函数、约束条件的数学表达式。以目标函数作为适应度函数用遗传算法进行求解。

目标函数：

$$Z = Z_1 + Z_2 + Z_3 \quad (3-1)$$

$$Z_1 = \sum_{p=1}^{P-1} \sum_{r=1}^R \sum_{i=1}^I a_p(i,r) \cdot x_{pi} \quad (3-2)$$

$$Z_2 = \sum_{p=2}^{P-1} \sum_{i=1}^I \sum_{s=1}^3 H_p(i,s) \cdot a_{ps} \quad (3-3)$$

$$Z_3 = \sum_{p=1}^{P-1} K \quad (3-4)$$

约束条件：

$$\sum_{i=1}^I j_i^D + \sum_{i=1}^I j_i^S = M_p \quad (3-5)$$

$$\sum_{i=1}^I j_i^S + \sum_{i=1}^I j_b^L = N_p \quad (3-6)$$

$$\sum_{p=1}^n a_p(i,r) - \sum_{p=1}^n R \leq 0 \quad (3-7)$$

$$\sum_{i=1}^I \sum_{s=1}^3 H_p(i,s) \cdot a_{ps} \leq \sum_{i=1}^I \sum_{s=1}^3 j_i^S \cdot m \quad (3-8)$$

$$\sum_{p=1}^P \sum_{j=1}^J D_{pj} - \sum_{p=2}^{P-1} \sum_{i=1}^I \sum_{s=1}^3 a_{ps} = \epsilon \quad (3-9)$$

$$\sum_{i=1}^n F_{pi} - \sum_{j=p+1}^J D_{pj} \geq 0 \quad (3-10)$$

式 (3-1) (3-2) (3-3) (3-4) 为目标函数，(3-1) 表示堆场、船舶整体倒箱和场桥多余移动次数之和最小，其中 k 表示倒箱与场桥移动次数的加权，即 1 次场桥来回移动时间对应多次倒箱操作时间；(3-2) 表示单独考虑堆场产生倒箱；(3-3) 单独考虑船舶产生的倒箱；公式 (3-4) 表示在出口箱区岸桥贝位间来回大距离移动的次数， k 越小越有利于船舶

的快速离港；公式（3-5）表示堆场中相应贝位或箱区装载的集装箱数量；公式（3-6）表示堆场中的相应贝位卸载的集装箱量；（3-7）表示该船舶在挂靠港口某贝位集装箱量不超过该贝位集装箱容量；（3-8）表示在一个港口中，产生的倒箱数不得大于该港口下舱盖上部堆箱数量；（3-9）表示所有港口配载的集装箱数量和与全航线装箱数量相等；（3-10）表示该船舶在当前挂靠港口卸载后的空箱位数量不能小于需要装载的箱量数。

3.4 本章小结

本章内容包括：集装箱船与堆场混合模型的提出和混合配载模型的建立。其中混合配载模型建立又包括：条件假设；参数和变量设计；优化目标及约束。为接下来算法设计提供一定的模型基础。

4 优化算法

集装箱船舶配载问题已被证明属于多目标组合优化问题（NP 问题），普通算法很难在多个约束中找到满意解，而遗传算法在解决 NP 问题上有独特的优势^[42-45]，故本文选用遗传算法来解决优化问题。遗传算法（GA）在 1975 年提出，本质上是染色体的进化问题。将当前种群的优秀染色体通过选择算子选择到子代染色体中，保存了种群的优良个体，再通过交叉算子对染色体进行重新组合，依据变异算子再进行染色体的突变，以得到更优个体。为了解决结果过早收敛的问题，本文在变异操作选用早已证明可以选用的单亲遗传算法^[30-34]；考虑到选取交叉点在同一处，不会产生非法染色体，交叉操作选用双亲遗传。

4.1 遗传算法流程

遗传算法具体算法流程如图 4.1。其基本机构主要包括四个部分：待解决问题的编码问题、适应度函数的设计问题、遗传操作选取和参数选取问题^[49]。

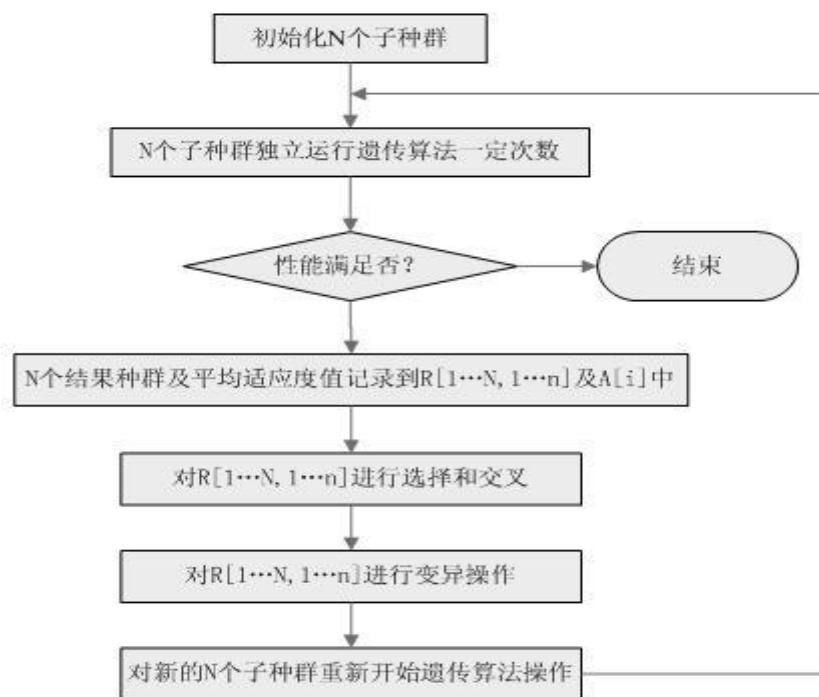


图 4.1 遗传算法流程

Figuer4.1 GA operating flow path

（1）编码。编码就是把待解决的问题通过编码基因组成染色体的方式转化为选取染色体的问题。编码的元素是基因，基因编码的方式有很多

种，有二进制编码（解决 01 背包问题）、互换编码（解决排序问题如旅行商问题）、树形编码、值编码^[29]。为了方便编码，本文采用值编码中的实数编码。

（2）适应度函数。适应度函数是考量个体适应环境能力大小的函数。适应度函数的选取可参考目标函数建立，一般设计为正负目标函数，为减小复杂程度，可以简化适应度函数。通过选取适应度值较大的个体繁殖进行遗传算子的操作后，选取适应度大的个体进行繁殖操作并选取适应度大的个体以淘汰适应度值低的个体。

（3）遗传操作。群体中个体进行优胜劣汰的操作算子叫遗传操作，包括：选择算子、交叉算子和变异算子^[29]。一般进行遗传操作都会选取不同的交叉概率和变异概率让染色体在相应概率下进行各自的遗传操作。

（4）运行参数。要根据实际情况确定不同的运行参数。本文选取种群规模： $M = 20-100$ ，选取终止进化代数： $T = 100-500$ ，选取变异概率 $P_m=0.2$ 交叉概率 $P_x=0.85$ 对染色体进行遗传操作。

4.2 初始解求解

初始解的求解过程为：

（1）根据运输矩阵随机产生各港口出口箱混堆矩阵以及随机产生的各港口取箱顺序，并记录相应的数据 $Z1$ 。

（2）根据产生的取箱顺序按配载原则配置上船并记录 $Z3$ 。配载原则为：随机选择贝位，按照取箱顺序依次对 6 个区间配载。舱盖下部区间始终保持着远的集装箱在下，近的集装箱再上；舱盖上部区间可任意配载，但必须保证不会导致舱盖下区间产生倒箱。若要装的箱数小于该贝位容纳，则都装在该贝位，否则，该贝位满，剩余的箱装在下一贝位，且在每个港口挂靠时，每个区间只能被配载 1 次。如图 4.4 为某贝位配载情况示例。

3	2	3	4	4	5
2	2	4	4	3	3
3	4	5	4	3	3
4	5	5	4	5	4

图 4.4 贝位配载示例

Figuer4.4 The example of bay loading

从图 4.4 中可以看到该贝位为分成 6 个区，为了保证舱盖下不产生倒箱，配载设计时第 1 行和第 2 行可以任意配载，但要保证第 3 行配载的集装箱要比第 2 行配载的集装箱要远，第 4 行配载的集装箱要比第 3 行的集装箱要远。因此本文倒箱主要发生在舱盖以上部分，记录下全航线的倒箱量，用 Z_2 表示。

(3) 输出配置计划、取箱顺序、倒箱量及场桥贝位间往复移动次数。

4.3 染色体编码

本文以实数编码方式进行编码。将混合配载编码设计成由出口箱区取箱顺序编码（染色体）影响船舶配载编码。编码主要分为：基于出口箱区混堆的染色体设计和堆场取箱顺序下的船舶配载设计两部分。为了编码方便，暂定船舶每个 bay 的分区容量一样。本文编码时，假设出口箱区堆高为 6，单船靠泊港口数为 5。

4.3.1 基于出口箱区混堆的染色体设计

首先，根据运输矩阵，随机构建堆场出口箱区堆箱状态 $E=[g_1, g_2, g_3, g_4]$ ， M 表示运输的所有集装箱量/6； a 、 b 、 c 、 d 对应相应港口目的箱，1 港口堆箱包含港口数值为 a 、 b 、 c 、 d 的集装箱；2 港口堆箱包含港口数值为 b 、 c 、 d 的集装箱；3 港口堆箱包含港口数值为 c 、 d 的集装箱；4 港口堆箱包含港口数值为 d 的集装箱，其中箱数不够构成矩阵用 0 补位。

$$E = \begin{bmatrix} b & \dots & c & c & \dots & b & c & \dots & c & d & \dots & d \\ c & \dots & c & c & \dots & d & d & \dots & c & d & \dots & d \\ c & \dots & a & b & \dots & d & c & \dots & d & d & \dots & 0 \\ d & \dots & b & b & \dots & c & c & \dots & d & d & \dots & 0 \\ a & \dots & d & d & \dots & b & d & \dots & d & d & \dots & 0 \\ a & \dots & c & b & \dots & d & d & \dots & d & d & \dots & 0 \end{bmatrix}_{6 \times M}$$

例如选取 E 中的 g_1 做示例，如图 4.2 中的出口箱区堆箱状态表示 1 号港口的堆箱。其中 0 表示该处没有堆箱。其堆存了 10 个在 2 号港口卸载的箱，15 个在 3 号港口卸载的箱，15 个在 4 号港口卸载的箱，15 个在 5 号港口卸载的箱。

$$E = \begin{bmatrix} 3 & 3 & 2 & 2 & 4 & 3 & 3 & 3 & 4 & 4 \\ 3 & 3 & 5 & 4 & 4 & 4 & 3 & 5 & 3 & 0 \\ 3 & 5 & 5 & 5 & 5 & 4 & 2 & 2 & 2 & 0 \\ 2 & 5 & 4 & 5 & 5 & 4 & 5 & 2 & 2 & 0 \\ 3 & 4 & 3 & 5 & 2 & 4 & 5 & 4 & 2 & 0 \\ 5 & 3 & 5 & 4 & 3 & 4 & 4 & 3 & 5 & 0 \end{bmatrix}$$

然后根据运输矩阵产生一个随机的取箱顺序 L (染色体), 即用取箱顺序 L 来取 E 中的箱。染色体中的 a 、 b 、 c 、 d 依旧表示相应港口的某个集装箱, $6 \times M$ 表示集装箱的总体运输量。

依旧选取港口 1 举例，如图 4.3 表示港口 1 的取箱顺序示例。其中 L 的长度为运输矩阵第一行的和，即在第一港口，船舶要装载的所有集装箱数量。其中数字代表各个港口的一个集装箱。

图 4.3 取箱顺序示例

4.3.2 堆场取箱顺序下的船舶配载设计

$$X = \begin{bmatrix} 0 & a & c & \dots & d & d & 0 \\ c & b & b & \dots & a & a & c \\ c & b & c & \dots & c & a & c \\ d & d & c & \dots & c & b & d \end{bmatrix}_{4 \times 84}$$

进行以下操作：

(1) 根据第一港口产生的取箱顺序，把集装箱随机分配到船舶各贝位中，得到船舶配载箱位分布 $R1$ ，即 $X=[R1]$ ；

(2) 到第 2 个港口时，先把 $R1$ 上编码为 2 的位置清空(即目的港为 2 的箱)，设置其编码为 0(卸船)，得到编码 $R11$ ，记录倒箱个数。然后在 $R11$ 上分配在港口 2 装船的集装箱时避免 $R1$ 中已经选取的区间，即下文提到的每个区间只被配载一次，构造出第 2 个港口出港时的箱位分布编码 $X=[R2]$ ，此时 $R2$ 已代替 $R1$ ；

(3) 继续在港口 3 至港口 4 执行操作 (2)，得到 $R3, R4$ ；

(4) 当 $R1$ 、 $R2$ 、 $R3$ 、 $R4$ 的到矩阵依次被代替得到最后的 X ：

$X=[R4]$ ；

X 为就是初始配载。

4.4 适应度函数

本文在适应度函数选取时并非考虑单个港口的倒箱量，而是考虑整条航线的倒箱以及整个航线的堆场的倒箱的总倒箱及整条航线的堆场中，场桥贝位间往复移动的次数最小。故本文适应度函数可设计为： $f(x) = C_x - Z$ ，其中 C_x 是无限大的正数， Z 为本文的目标函数。即本文 Z 分别为堆场倒箱和船舶总倒箱与场桥贝位间往复移动次数之和。

4.5 选择操作

本文采用轮盘赌的操作方法^[50-52]，来选取适应度值大的个体。例如假设一个规模为 5 的种群有个体 $a1, a2, a3, a4, a5$ ，其适应度函数值分别为 1, 2, 3, 4, 5。那么可以计算出种群适应度函数值的和 $1+2+3+4+5=15$ 。个体 $a1$ 被选中进入下一代的概率 $1/15$ 就是自身在种群适应度函数和值的占比，同理可得 $a2, a3, a4, a5$ 的

被选取进入下一代的概率分别为 $2/15$ 、 $3/15$ 、 $4/15$ 、 $5/15$ 。按照这样得到的概率进行选择个体，可以发现适应度值越大的个体越有优势。

本文的具体操作就是选取 50 个个体，通过记录其不同的适应度值进行上述操作。

4.6 遗传算子

本文的遗传算子^[53-54]包括：在一条染色体上进行本变异（单亲遗传）、在两条染色体间交叉两部分（双亲遗传）。其中，因两条染色体交叉操作的交叉点选取位置一样，故不会产生不合法的染色体，简化了遗传操作，提高运算效率。

4.6.1 交叉操作

本文采用交叉操作方法的具体过程如下：

(1) 随机产生一个小数 r 并与设定参数进行对比，如果 r 小于设定的参数则存放需要交叉的染色体序号，并记录需要交叉染色体的个数，若染色体个数为奇数，则染色体少记录一条转成偶数，若是偶数无需变动；

(2) 在染色体长度范围里随机产生交叉的位置并按升序排列，选择最前的位置 $a11$ 作为交叉点；

(3) 选取两个父代染色体的前 $a11$ 段基因，进行交换基因的互换操作，得到两个子代染色体，本文在设计编码时为了方便，故意设计成交叉点相同，避免染色体的产生；

(4) 记录此时变异后染色体倒箱，并与父代其它染色体的倒箱进行比较，以选取倒箱最小染色体。

4.6.2 变异操作

本文采用选取群体中的某些个体，对其进行基因突变操作，此次基因突变可以说整条染色体的任何位置，具体过程如下：

(1) 随机产生一个小数 rr 并与设定的参数进行比较，如果小数 rr 小于设定参数则转(2)，否则转(5)；

(2) 随机产生 2 个自然数 $r1$ 和 $r2$ 记录其位置， $r1$ 和 $r2$ 表示染色体的变异位；

(3) 交换染色体的第 $r1$ 列和第 $r2$ 列；

(4) 更新染色体；

(5) 记录此时变异后染色体倒箱，并与父代其它染色体的倒箱进行比较，以选取倒箱最小染色体。

4.7 数值模拟

本文选用一个算例数值模拟来验证模型的可行性及实用性。

算例满足以下假设条件：1) 船舶各贝位均为等截面贝位，即贝位的形状及其载箱量一致；2) 一个贝位划分为 4 个舱位，每个舱位甲板下有 4 个堆垛，每个堆垛最多堆放 9 个箱；每个舱位甲板上共可堆放 20 个标箱。船舶的各贝位载箱量信息如表 4.1 所示：

表 4.1 各港口船舶各贝位的载箱量
Tab 4.1 Container ship capacity of each bay

船舶 贝位		2	6	10	14	18	22	26	30	34	38	42
港 1	装 箱	0	0	20	20	30	10	10	10	10	0	30
	卸 箱	0	0	0	20	20	0	0	0	0	0	0
港 2	装 箱	10	10	10	20	10	10	10	10	10	0	0
	卸 箱	0	0	0	0	0	10	20	0	0	0	0
港 3	装 箱	0	20	0	0	0	10	18	0	10	10	0
	卸 箱	0	0	13	0	10	10	10	0	0	0	0
港 4	装 箱	10	0	0	0	8	10	10	0	0	0	0
	卸 箱	20	30	13	20	18	20	18	20	30	10	30
港 5	装 箱	0	0	0	0	0	0	0	0	0	0	0
	卸 箱	0	0	0	0	0	0	0	0	0	0	0

表 4.2 装箱矩阵
Tab 4.2 The matrix of packing

	1	2	3	4	5	装载总箱数
1	--	40	25	35	40	140
2	--	--	30	40	30	100
3	--	--	--	43	25	68
4	--	--	--	--	38	38

假设船舶由始发港 $P=1$ 开始装箱，途经 3 个挂靠港，到第 5 个港口卸载完所有集装箱，根据表 4.1 得到装箱矩阵信息，如表 4.2 所示。

由装载矩阵随机产生堆场出口箱区混堆信息，如表 4.3。混堆矩阵是一个 $6 \times N$ 的矩阵，表示各港口的堆箱状态。

表 4.3 各港口出口箱区混堆情况
Table 4.3 Each port exports box area mixing pile

g_1	2	3	5	4	5	4	4	4	3	5	4	4	4	3	2	2	2	4	2	2	4	3	5	5
	3	3	2	5	2	3	4	5	4	2	2	4	3	4	4	5	4	2	5	3	5	4	4	2
	5	2	4	2	4	4	4	5	3	5	3	2	2	2	5	4	3	3	4	4	2	5	5	0
	5	2	5	2	2	2	2	3	3	2	3	3	3	3	5	5	4	5	5	5	2	3	4	0
	5	2	3	4	5	4	2	2	3	4	5	2	3	5	5	4	3	2	2	2	5	5	2	0
	5	5	2	2	2	3	2	5	4	2	4	5	2	4	4	5	5	4	5	2	5	5	5	0
g_2	4	5	4	3	4	4	5	4	3	4	4	5	4	4	4	5	5							
	4	3	3	3	5	3	4	5	3	3	4	4	4	4	4	5	5							
	5	5	4	3	5	3	5	5	3	4	5	3	4	4	4	3	5							
	5	3	5	3	5	4	4	5	4	5	4	3	4	4	3	3	3							
	5	5	3	3	3	3	4	3	3	5	4	5	4	5	5	4	0							
	4	5	4	3	3	3	5	5	4	3	4	4	3	4	4	4	0							

$$g_4 = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & 0 \end{bmatrix}$$

1. 解出岸桥贝位间来回移动次数为 4 次;
2. 解出试验港口最优取箱顺序 **Result1** 如表 4.4, 取箱顺序是一个 $1 \times M$ 的向量, 其中 **M** 为全航线取箱总量;

Result1=[4, 2, 3, 4, 4, 5, 0, 2, 4, 3, 5, 2, 4, 3, 4, 5, 2, 5, 3, 5, 3, 2, 5, 3, 4, 4,
2, 5, 2, 5, 5, 4, 2, 3, 5, 4, 4, 2, 5, 2, 2, 4, 4, 2, 5, 5, 3, 2, 0, 5, 5, 4, 2,
3, 4, 2, 3, 2, 2, 2, 4, 5, 4, 2, 3, 3, 4, 4, 5, 4, 2, 5, 4, 2, 5, 3, 3, 4, 3, 2
5, 3, 5, 2, 5, 2, 0, 4, 4, 0, 5, 2, 5, 5, 2, 2, 5, 2, 2, 5, 5, 2, 3, 4, 4, 3, 3,
5, 5, 3, 4, 2, 4, 3, 4, 2, 5, 4, 5, 3, 5, 2, 5, 5, 2, 2, 4, 2, 2, 4, 2, 4, 4, 4,
5, 2, 2, 4, 5, 3, 5, 4, 3, 5, 5, 3, 4, 4, 4, 3, 3, 5, 4, 4, 4, 3, 5, 4, 5, 0, 4,
5, 5, 4, 5, 4, 5, 4, 4, 5, 4, 4, 4, 4, 4, 3, 3, 4, 4, 5, 3, 3, 5, 5, 4, 4, 3, 4,
4, 5, 3, 4, 4, 4, 3, 4, 3, 4, 4, 3, 3, 5, 3, 3, 3, 5, 3, 3, 4, 3, 4, 5, 3, 5, 4,
3, 3, 5, 4, 3, 4, 4, 3, 5, 5, 4, 5, 5, 5, 5, 4, 5, 5, 5, 5, 5, 3, 3, 3, 4, 5, 3,
5, 0, 3, 4, 4, 3, 5, 4, 5, 4, 4, 5, 4, 4, 5, 4, 4, 4, 4, 4, 4, 4, 4, 5, 4, 5,
4, 5, 5, 4, 4, 4, 4, 0, 5, 4, 4, 5, 5, 4, 4, 0, 5, 5, 5, 4, 4, 5, 4, 4, 5, 4, 4,
5, 4, 4, 5, 5, 4, 4, 4, 4, 5, 5, 5, 4, 5, 4, 4, 0, 4, 4, 0, 5, 4, 5, 4, 5, 5, 0,
5, 0, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 0, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 0, 5]

- 36 -

表 4.5 最优配载矩阵
Table 4.5 The optimal loading matrix

ord = 10, 6, 14, 8, 3, 4, 9, 7, 2, 11, 13, 12, 1, 5

rrc =

2, 5, 2, 4, 3, 5, 2, 2, 3, 2, 2, 4, 0, 0, 0, 0, 0, 0, 3
2, 3, 3, 3, 2, 4, 2, 3, 2, 2, 4, 4, 0, 0, 0, 0, 0, 0, 3
3, 5, 4, 5, 2, 4, 5, 4, 2, 3, 5, 5, 5, 2, 4, 3, 2, 4, 4
4, 5, 4, 5, 3, 5, 5, 5, 2, 4, 5, 5, 5, 4, 4, 5, 3, 4, 5
2, 5, 5, 2, 2, 2, 2, 5, 4, 4, 2, 4, 5, 5, 4, 5, 4, 4, 4
2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 4, 3, 3, 4, 3, 4, 4, 4, 4
4, 2, 5, 3, 3, 4, 3, 5, 3, 2, 4, 3, 3, 4, 4, 5, 5, 4, 4
5, 2, 5, 5, 5, 5, 4, 5, 5, 3, 5, 5, 4, 5, 4, 5, 5, 5, 5
3, 5, 3, 4, 4, 4, 3, 4, 5, 3, 4, 4, 5, 4, 5, 3, 4, 5, 4
3, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 5, 4, 3, 3, 4, 4, 4
4, 3, 4, 5, 3, 3, 3, 4, 3, 4, 4, 5, 5, 5, 5, 3, 4, 4, 4
5, 5, 4, 5, 4, 4, 4, 5, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 5
5, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 0, 0, 0, 0
4, 4, 4, 4, 4, 4, 4, 5, 4, 5, 5, 5, 4, 5, 5, 0, 0, 0, 0
5, 4, 5, 4, 5, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 0, 0, 5, 5
5, 4, 5, 5, 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5
0, 0
0, 0
5, 5
5, 5

4. 解出试验船舶与堆场混合倒箱最优解迭代图，如图 4.5。由图可知，倒箱数随着迭代次数的增加而渐渐减小，当迭代到 92 次时，倒箱次数为 146 次。

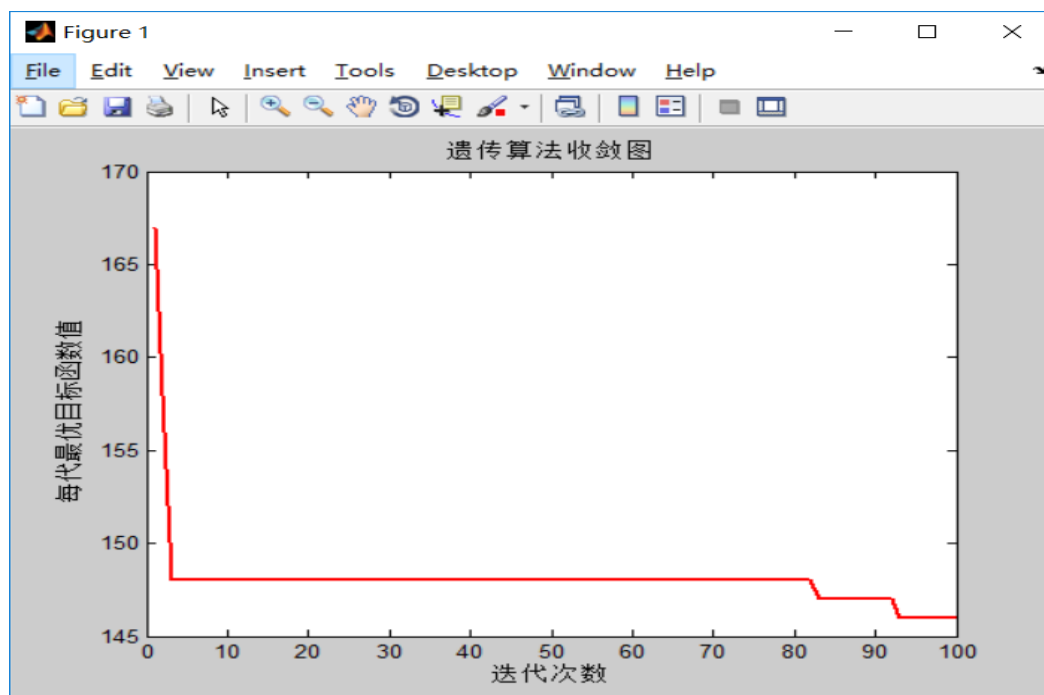


图 4.5 遗传算法迭代图

Figure 4.5 Shift of GA

相比较传统倒箱（考虑先取最远港口集装箱，再取较近港口集装箱，最后取最近集装箱来计录倒箱）165，减少 11.51% 的倒箱率。本文设计方法致使场桥多移动 4 次，根据场桥移动与倒箱量之间的权重关系，其计算结果依旧满足减少倒箱量。

4.8 本章小结

本章根据提出的模型介绍了优化该模型的方法：遗传算法。介绍遗传算法的流程，最后根据遗传算法的流程作出合适的算法设计。算法设计又包括：染色体编码；适应度函数的选择；选择操作；变异。通过数值模拟验证本文选用的设计方案的有效性，为下章应用做基础。

5 工程应用

本文为了理论联系实际，将理论付诸于实际，选用宁波港北仑港吉码公司为试点单位，进行工程应用计算。下面简单介绍港吉公司情况。

5.1SPARCE 系统简介

如图 5.1 为北仑港现用的 SPARCE 系统。它是一款时时化的软件，用来对码头运营的管理，本文只做简单介绍。它的功能强大，不仅可以优化设备调度还可以时时反应堆场使用情况。

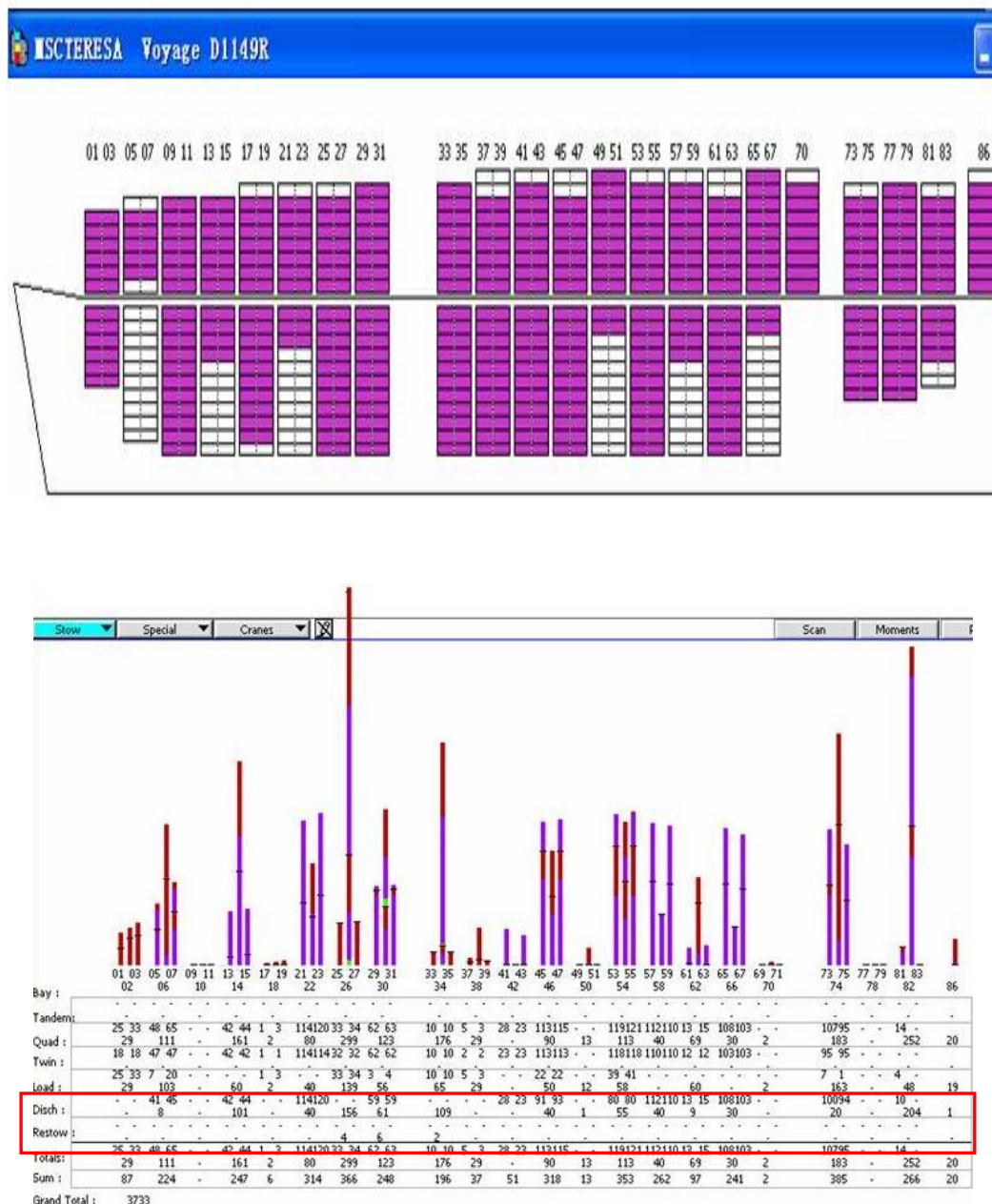


图 5.1 SPARCE 系统船舶界面图

Figure 5.1 SPARCE system interface diagram of the ship

5.2 实例数据

本文考虑单条船经过多个港口的配载，很难追踪到实际数据；从港吉公司采集到数据仅仅是单条船某时刻的数据且非起始港数据，故需要对数据进行处理。处理思路：选取港吉公司同条船的不同时刻作为各港口的装卸情况。选取前 5 个港口的数据处理结果如表 5.1 贝位内对应的进/出口箱量。

表 5.1 贝位内对应的进/出口箱量

Table.5.1 Amount of import and export containers in each bay of a yard

unit: FEU

贝位号	1	2	3	4	5	6	7	8	9	10
出口箱量	0	0	13	10	11	12	13	0	13	0
进口箱量	0	0	0	0	0	0	0	0	0	0
贝位号	11	12	13	14						
出口箱量	11	10	12	15						
进口箱量	0	0	0	0						
贝位号	1	2	3	4	5	6	7	8	9	10
出口箱量	13	7	0	0	0	7	0	12	0	11
进口箱量	0	0	5	0	0	7	5	0	5	0
贝位号	11	12	13	14						
出口箱量	6	0	0	6						
进口箱量	5	0	0	8						
贝位号	1	2	3	4	5	6	7	8	9	10
出口箱量	0	4	0	12	7	0	13	0	7	6
进口箱量	0	0	0	10	4	0	8	0	8	5
贝位号	11	12	13	14						
出口箱量	0	7	0	0						
进口箱量	0	4	0	0						
贝位号	1	2	3	4	5	6	7	8	9	10
出口箱量	0	0	0	4	13	0	0	0	0	14
进口箱量	11	5	0	5	7	12	7	0	0	12
贝位号	11	12	13	14						
出口箱量	10	13	0	0						
进口箱量	5	0	3	13						

处理数据得到各港口的装载情况如表 5.2。

表 5.2 各港口进行操作的集装箱量
Table 5.2 Amount of import and export container ports

unit: FEU

港口 \ 集装箱量		1	2	3	4	5	6	...
1	进口箱量	0	0	0	0	0	0	...
	出口箱量	0	35	19	30	26	15	...
2	进口箱量	0	35	0	0	0	0	...
	出口箱量	0	0	20	20	22	7	...
3	进口箱量	0	0	39	0	0	0	...
	出口箱量	0	0	0	30	26	17	...
4	进口箱量	0	0	0	80	0	0	...
	出口箱量	0	0	0	0	13	9
5	进口箱量	0	0	0	0	87	0	...
	出口箱量	0	0	0	0	0	15	...
6	进口箱量	0	0	0	0	0	63	...
	出口箱量	0	0	0	0	0	0	...
· · ·	进口箱量
	出口箱量

根据以上数据，得到具体的装箱矩阵，如表 5.3。

表 5.3 装箱矩阵
Table 5.3 The matrix of packing

		1	2	3	4	5	装载总箱数
起 始 港	1	--	35	19	30	26	110
	2	--	--	20	20	22	62
	3	--	--	--	30	26	56
	4	--	--	--	--	13	13

考虑到此次装载的集装箱数量 相对较少，此次实例应用，将堆场堆箱高度设定为 4。得到装箱矩阵后可以按照第 4 章数值模拟进行操作。

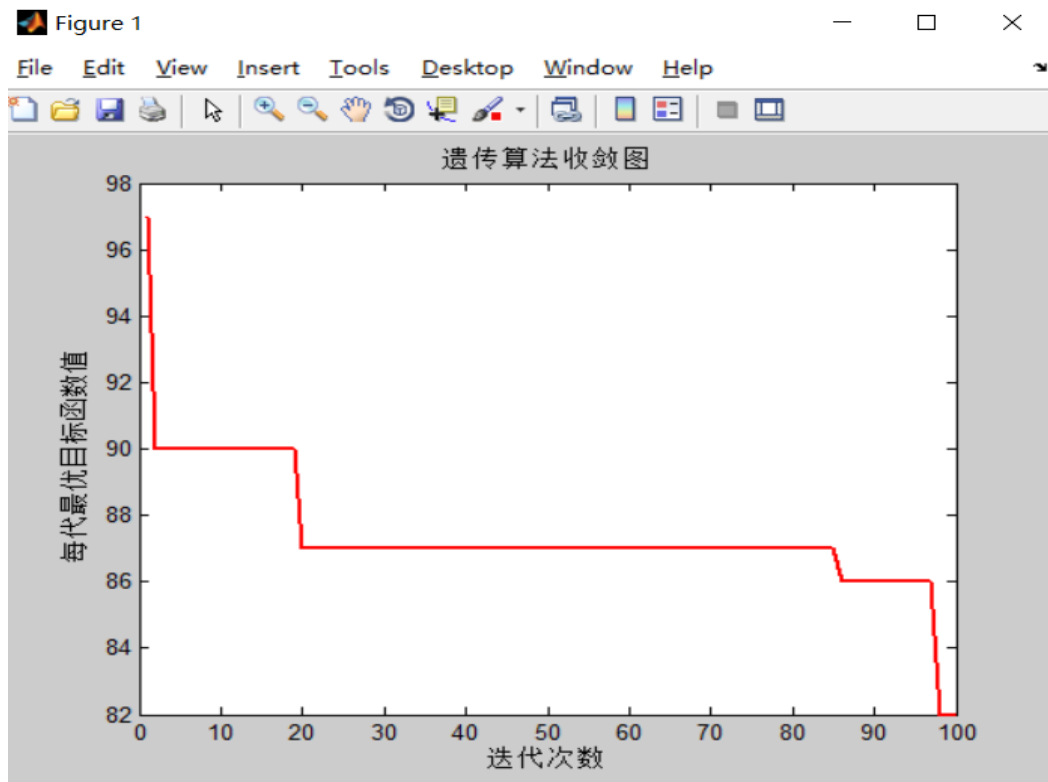


图 5.2 遗传算法迭代图

Figure 5.2 Shift of GA

初始种群规模 50，迭代代数 100 代， $P_m=0.2$, $P_x=0.85$ 的变异和交叉概率，

计算到第 97 代时求得最优解。得到结果为：倒箱 82 次，取箱时岸桥贝位间往复移动 0 次数，如图 5.2 为倒箱次数迭代图。相比较于 SPARCE 系统取箱得到的 $Z=89$ （仅考虑堆场的倒箱，还并未考虑船舶倒箱）少 7 个倒箱，减少倒箱率 7.86%。

6 结论与展望

6.1 研究结论与创新

为了改变传统的配载方式，研究减少堆场倒箱、减少岸桥移动时间、减少船舶产生倒箱的方法，从而减少船舶进出港滞留时间。本文通过对现实的船舶配载以及相关文献的研究，结合如今集装箱船舶的配载模式，深入分析了装船作业时倒箱产生的原因，提出了考虑堆场出口箱混堆情况下船舶与堆场混合的配载方案。

本文所做的研究成果与创新点如下：

（1）模型的创新。本文混合配载考虑到堆场未预倒箱的情况。基于出口箱混堆情况，建立以全航线船舶倒箱量与堆场倒箱量最少及场桥贝位间往复移动次数最小数学模型；

（2）权重的设定创新。引入倒箱量与场桥取箱往复移动次数的权重¹，并通过数学推理推导出具体大小；

（3）堆场取箱方法创新。采用扫描取箱法，来确定集装箱的提取顺序；

（4）配载设计创新。设计满足船舶舱盖下部不产生倒箱的发生，大大降低船舶因舱盖下倒箱造成的长时间滞留，同时舱盖上部可以产生混 bay 来预调船舶安全性能；

（5）数据处理方法创新。采集实际数据验算算法时，因单船运动轨迹复杂难以追踪，本文以单船的不同时间段到港的数据作为单船在不同港口的装载数据。

6.2 研究展望

本文的研究仍有很多待改善之处：

（1）本文主要研究的是单条集装箱船舶与堆场的混合配载问题，但仅仅是考虑倒箱量和岸桥贝位间往复移动次数的问题。本文中没有考虑堆场作业效率、场桥作业时间、装箱时岸桥的来回移动等，后期可以逐个考虑。

（2）本文假设产生的混贝位能调节船舶安全性能，却没有来得及带入各集装箱重量验算导致船舶倒箱量相对增加，后期可以通过验算，寻求合理的混贝产生来减少倒箱。

（3）本文在堆场取箱方面及数据采集方面有待改善。编码时运用扫描法取箱，这样有可能产生多次场桥的移动，在后期可以作为约束；数据采集难以得到单船全航线数据，有待改善。

参考文献

- [1] Nguyen S, Zhang M, Johnston M, Chen Tan K. Hybrid evolutionary computation methods for quay crane scheduling problems[J]. Computers & Operations Research. 2013. 40(8): 2083-2093.
- [2] Kaveshgar N, Huyuh N, Rahimian S. An efficient genetic algorithm for solving the quay crane scheduling problem [J]. Expert Systems with Applications. 2012. 39(18): 13108-13117.
- [3] Zhao N, Mi W J. Robust approach in stowage planning at container terminals[C]. Proceedings of the 4th International Conference on Intelligent Logistics Systems, 2008: 191-204.
- [4] Akio I, Dazuya S, Etsuo N, et al. Multi-objective simultaneous stowage and load planning for a container ship with container re-handle in yard stacks. European Journal of Operational Research, 2006. 171(2): 373-389.
- [5] Avriel, M., Penn, M., Exact and approximate solutions of the container ship stowage problem[C]. Computers & Industrial Engineering 1993. 25, 271-274.
- [6] Imai A, Nishimura E, Sasaki K, Papadimitriou S. Solution comparisons of algorithm for the container ship loading problem[C]. Proceeding of the International Conference on Shipping Technology and Environment, available on CDROM. 2001.
- [7] Ambrosino, D., Sciomachen, A., Tanfani, E., Stowing a container ship: The master bay plan problem. Transportation Research A 2003. 38(2), 81-99.
- [8] O Dubrovsky, G Levitin, M Penn. A genetic algorithm with a compact solution encoding for the container ship stowage problem[J]. Journal of Heuristics, 2002, 8(6): 585-599.
- [9] 朱雯. 集装箱船多港配载优化研究[D]. 武汉理工大学, 2013, 57(6): 8-15
- [10] 张维英. 集装箱船全航线配载智能优化研究[D]. 大连: 大连理工大学, 2005.
- [11] 张维英. 林焰. 集装箱船全航线 Bay 位排箱优化模型[J]. 上海交通大学学报, 2007. 41(2). 199-204.
- [12] 张维英. 林焰. 集装箱船全航线预配优化模型与算法研究[J]. 大连理工大学学报, 2008. 48(5). 673-678.
- [13] 张维英. 基于指派问题的 Bay 位排箱优化模型与算法[J]. 大连理工大学学报, 2011. 51(1). 61-67.
- [14] 卫家骏. 一种集装箱船配载问题改进算法探讨[J]. 重庆交通大学学报(自然科学版), 2009. 28(5). 969-972.
- [15] 卫家骏. 集装箱船舶智能配载研究[D]. 大连海事大学, 2012.
- [16] 孙晓雅, 林焰. 集装箱船多港 bay 位排箱的优化方法[J]. 大连海事大学学报, 2011. 37(1). 75-79
- [17] 靳志宏, 兰辉. 基于现实约束的集装箱装船顺序优化[J]. 大连海事大学学报,

2011.37(1).71-74

- [18] 曲目, 计明军. 集装箱船舶智能配载优化问题研究[D]. 大连海事大学, 2011.
- [19] 王君红. 基于网格化的集装箱船舶全航线优化配载问题研究[D]. 大连海事大学, 2013
- [20] 祝慧灵, 计明军. 集装箱船舶全航线配载优化模型与改进遗传算法[J]. 交通运输工程学报, 2014.14(5).59-67.
- [21] 陈庆伟, 王继荣. 集装箱堆存出口箱堆存模型及算法[J]. 物流科技, 2007,(7)106-108.
- [22] 卫家骏. 出口集装箱堆场位置的优化[J]. 重庆交通大学学报, 2010,29(3):470-473.
- [23] 周鹏飞, 方波. 动态环境下集装箱码头堆场出口箱箱位分配建模预算法研究[J], 2011,26(10):1512-1576
- [24] 刘庆, 集装箱码头的船舶配载与堆场规划. 物流工程与管理[D], 2012.
- [25] 王秀臣, 出口箱随机入港下码头堆场与船舶积载协调优化[D], 大连海事大学, 2013.
- [26] 李茂军, 童调生, 罗隆福. 单亲遗传算法及其应用研究[J]. 湖南大学学报, 1998.25(6):56-59.
- [27] 李茂军, 邱丽芳. 单亲遗传算法的计算效率分析[J]. 长沙电力学院学报(自然科学版) 1999.14(1):48-50.
- [28] 李茂军, 童调生. 单亲遗传算法及其全局收敛性分析[J]. 自动化学报, 1999.25(1):68-72.
- [29] 李茂军, 童调生. 单亲遗传算法编码方式的研究[J]. 长沙电力学院学报(自然科学版) 2000.15(3):11-13.
- [30] 陈慧琴, 刘刚. 用整数编码的单亲遗传算法求解组合优化问题[J]. 武汉理工大学学报(交通科学与工程版), 2003.7(2):241-243.
- [31] 肖晓明, 旷东林, 蔡自兴. 单亲遗传算法种群初始化方法分析[J]. 电脑与信息技术, 2009,17(4): 1-3.
- [32] 邓欣, 朱征宇, 曾凡超. 一种多车场车辆路径问题的单亲遗传算法[J]. 交通与计算机, 2007,25(1): 35-47.
- [33] 王敏, 李铁克. 改进的单亲遗传算法求解倒垛问题[J]. 工业工程与管理, 2009,14(4):67-71.
- [34] 邓欣, 朱征宇, 曾凡超. 一种多车场车辆路径问题的单亲遗传算法[J]. 交通与计算机, 2007,25(1): 35-47.
- [35] 胡燕海, 严隽琪, 叶飞帆. 基于遗传算法的混合流水车间构建方法[J]. 中国机械工程, 2005,16(10):888-891.
- [36] 何赞燕, 胡燕海. 基于舱位协同作业的集装箱同步装卸方法[J]. 宁波大学学报, 2012,25(3): 105-108
- [37] 王启友. 集装箱船的单元块配载[J]. 集装箱化. 2007.8:26-29.
- [38] 陶其钧. 集装箱堆场机械选型配置研究[J]. 港口装卸, 2004, 141(1):1-6.

- [39] 王莉莉. 集装箱装船顺序优化模型及遗传算法[D].大连理工大学,2008,44(11):2-11.
- [40] 杨晓东. 仿真技术在集装箱船舶配载问题中的应用研究[J].中国科技博览,2010(26):314-315.
- [41]Kun Li.Modelling and Tabu search heuristic for solving container stowage planning problem[J]. Computers & Industrial Engineering,2012,24:2676-2680.
- [42] 黎明. 集装箱码头箱位分配与装船顺序研究[D].鲁东大学,2012.
- [43]康海贵, 刘艳, 周鹏飞. 基于混堆的集装箱堆场动态箱位分配研究[J]. 水运工程, 2009,8(430): 73-78.
- [44] 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, 2009,29(8): 185-192.
- [45] 曾庆成, 孙丽敏, 杨忠振. 集装箱码头装卸混合调度模型[J]. 哈尔滨工程大学学报, 2009,41(10):287-289.
- [46] 朱明华. 多工况集装箱船配载与堆场翻箱优化研究[D].上海交通大学,2011.7.
- [47] 朱明华. 集装箱船舶智能配载研究[D].大连海事大学, 2012.
- [48] 朱明华, 范秀敏. 集装箱装船顺序问题的启发式算法研究[J].中国机械工程, 2010, [12]
[12]
- [49] 王雅芬. 集装箱船舶配载智能决策研究[D].武汉科技大学.201521(9):1066-1070.
- [50] 孙俊清, 刘凤连. 考虑船舶稳定性的多港口集装箱配载问题[J].计算机工程与应用,2012,48(32):236-242.
- [51] 卢萍. 基于稳定约束的集装箱配载计划启发式方法研究[D].天津大学,2008.
- [52] 史宗耀, 崔广才. 混合遗传算法在集装箱船舶配载中的应用[D]. 长春理工大学, 2010,62(11).
- [54] 杨晓东. 仿真技术在集装箱船舶配载问题中的应用研究[J].中国科技博览,2010(26):314-315.

致谢

回想硕士研究生生活，能有机会静下心来研究某个领域，尤其是集装箱码头方面的调度问题，让我受益良多。在本课题研究的整个过程中，我要深深地感谢以下对我有过支持和帮助的人：

论文能够顺利完成，首先我要感谢我的导师胡燕海老师，从选题到论文即将完成的这段时间中，给予了莫大的帮助。在我课题研究出现困境时，胡老师总能提出许多宝贵意见，在他的指导下，让我的论文研究道路越来越明朗。不仅在科研上，在校期间我遇到的学习、生活、以及工作的问题，胡老师都会细心帮助我，给我提出建设性意见。

其次，我要感谢我们课题组的师姐、师兄及师弟们，课题进展顺利离不开他们在技术及资料提供上提供的大力帮助。

再次，我要感谢两年多时间来，我们海运 2014 级研究生班级的每位同学，你们是我好的伙伴，无论学习还是生活上都给我提供了很多帮助，相亲相爱，互助向上。

最后，我要感谢我的家人，让我有一个安静学习的机会，能顺利完成硕士研究生阶段的科研任务，离不开他们的经济与精神上的支持。

附录：

GA.m（遗传算法主程序）

```
function []=GA()
clc
clear
global G g1 g2 g3 g4 Bay
T=[0,40,25,35 40;
    0,0,30,40 30;
    0,0,0,43 25;
    0,0,0,0 38];
n=4;%假设堆场高为 4
Bay=[37    32    40    30    31    33    29    28    32    32
35    32 44 12];%bay 位容量
G=5;%港口总数
pop=50;%染色体数目
MaxGen=100;%迭代次数
z=0;%传统倒箱记录
px=0.85;%交叉概率
pm=0.2;%变异概率
bay=length(Bay);
%%%随机产生各堆场堆箱
e=duixiangshunxu(n,T);
[x,y]=size(e);
%%%传统方法计算倒箱
for h=1:y
    C=e(:,h);
    if length(find(C>0))==4
        D=find(C>0);
        if (C(D(3))<C(D(1))&&C(D(1))<C(D(2))&&C(D(2))<C(D(4)))|
(C(D(3))<C(D(2))&&C(D(2))<C(D(1))&&C(D(1))<C(D(4)))|(C(D(2))<C(D(1))&&C(D(
1))<C(D(3))&&C(D(3))<C(D(4)))|(C(D(2))<C(D(3))&&C(D(3))<C(D(1))&&C(D(1))<C
(D(4)))| (C(D(1))<C(D(2))&&C(D(2))<C(D(3))&&C(D(3))<C(D(4)))|
(C(D(1))<C(D(3))&&C(D(3))<C(D(2))&&C(D(2))<C(D(4)))|
```

```

(C(D(1))==C(D(2))&&C(D(1))<C(D(3))&&C(D(3))<C(D(4)))|
(C(D(1))==C(D(2))&&C(D(2))==C(D(3))&&C(D(3))<C(D(4)))|
(C(D(1))==C(D(2))&&C(D(3))<C(D(1))&&C(D(1))<C(D(4)))|(C(D(1))==C(D(3))&&C(
D(1))<C(D(2))&&C(D(2))<C(D(4)))|(C(D(1))==C(D(3))&&C(D(2))<C(D(1))&&C(D(1)
)<C(D(4)))|          (C(D(3))==C(D(2))&&C(D(1))<C(D(2))&&C(D(2))<C(D(4)))|
(C(D(3))==C(D(2))&&C(D(3))<C(D(1))&&C(D(1))<C(D(4)))

    z=z+3;

end

if
(C(D(4))<C(D(2))&&C(D(2))<C(D(3))&&C(D(3))<C(D(1)))|(C(D(4))<C(D(3))&&C(D(
3))<C(D(1))&&C(D(1))<C(D(2)))|(C(D(3))<C(D(4))&&C(D(4))<C(D(2))&&C(D(2))<C
(D(1)))|(C(D(2))<C(D(4))&&C(D(4))<C(D(3))&&C(D(3))<C(D(1)))|(C(D(1))<C(D(4))
&&C(D(4))<C(D(3))&&C(D(3))<C(D(2)))|(C(D(1))==C(D(2))&&C(D(3))<C(D(4))&&
C(D(4))<C(D(1)))|(C(D(1))==C(D(3))&&C(D(4))<C(D(1))&&C(D(1))<C(D(3)))|(C(D(
1))<C(D(3))&&C(D(4))<C(D(2))&&C(D(2))<C(D(1)))
|C(D(1))==C(D(4))&&C(D(1))<C(D(3))&&C(D(3))<C(D(2)))|
(C(D(1))==C(D(3))&&C(D(3))<C(D(4))&&C(D(1))<C(D(2)))|
(C(D(1))==C(D(2))&&C(D(3))<C(D(1))&&C(D(1))<C(D(4)))|
(C(D(1))==C(D(4))&&C(D(2))<C(D(3))&&C(D(1))<C(D(2)))|
(C(D(1))==C(D(3))&&C(D(2))<C(D(4))&&C(D(1))>C(D(2)))|
(C(D(2))<C(D(4))&&C(D(2))<C(D(3))&&C(D(3))<C(D(1)))|
(C(D(4))<C(D(2))&&C(D(3))<C(D(2))&&C(D(2))<C(D(1)))|
(C(D(1))<C(D(3))&&C(D(2))<C(D(4))&&C(D(4))<C(D(1)))|
(C(D(1))<C(D(2))&&C(D(2))<C(D(4))&&C(D(3))<C(D(4)))|
(C(D(3))<C(D(2))&&C(D(4))<C(D(1))&&C(D(1))<C(D(2)))|
(C(D(4))<C(D(3))&&C(D(1))<C(D(3))&&C(D(3))<C(D(2)))|
(C(D(4))<C(D(3))&&C(D(2))<C(D(3))&&C(D(3))<C(D(1)))|
(C(D(4))<C(D(2))&&C(D(2))<C(D(3))&&C(D(1))<C(D(3)))|
(C(D(3))<C(D(2))&&C(D(1))<C(D(2))&&C(D(2))<C(D(4)))|
(C(D(4))<C(D(1))&&C(D(2))<C(D(3))&&C(D(1))<C(D(3)))

    z=z+1;

end

if
(C(D(4))<C(D(1))&&C(D(1))<C(D(2))&&C(D(2))<C(D(3)))|(C(D(4))<C(D(1))&&C(D(

```

```

1))<C(D(3))&&C(D(3))<C(D(2))|(C(D(4))<C(D(2))&&C(D(2))<C(D(1))&&C(D(1))<C
(D(3))|(C(D(3))<C(D(1))&&C(D(1))<C(D(4))&&C(D(4))<C(D(2))|(C(D(3))<C(D(2))
&&C(D(2))<C(D(4))&&C(D(4))<C(D(1))|(C(D(3))<C(D(4))&&C(D(4))<C(D(1))&&C
(D(1))<C(D(2))|(C(D(2))<C(D(1))&&C(D(1))<C(D(4))&&C(D(4))<C(D(3))|(C(D(2))
<C(D(3))&&C(D(3))<C(D(4))&&C(D(4))<C(D(1))|(C(D(2))<C(D(4))&&C(D(4))<C(D
(1))&&C(D(1))<C(D(3))|(C(D(1))<C(D(2))&&C(D(2))<C(D(4))&&C(D(4))<C(D(3))|(
C(D(1))<C(D(3))&&C(D(3))<C(D(4))&&C(D(4))<C(D(2))|(C(D(1))<C(D(4))&&C(D(
4))<C(D(2))&&C(D(2))<C(D(3))|(C(D(1))==C(D(2))&&C(D(1))<C(D(4))&&C(D(4))<
C(D(3))|(C(D(1))==C(D(2))&&C(D(4))<C(D(1))&&C(D(1))<C(D(3))|(C(D(1))==C(D
(3))&&C(D(1))<C(D(4))&&C(D(4))<C(D(2))|(C(D(1))==C(D(4))&&C(D(1))<C(D(2))
&&C(D(2))<C(D(3))|(C(D(1))==C(D(4))&&C(D(2))<C(D(1))&&C(D(1))<C(D(3))|(C
(D(1))==C(D(4))&&C(D(2))<C(D(3))&&C(D(3))<C(D(1))|(C(D(1))==C(D(4))&&C(D
(3))<C(D(1))&&C(D(1))<C(D(2))|(C(D(1))==C(D(4))&&C(D(3))<C(D(2))&&C(D(2))
<C(D(1))|
(C(D(1))==C(D(2))&&C(D(2))==C(D(4))&&C(D(1))<C(D(3))|
(C(D(1))==C(D(2))&&C(D(3))==C(D(4))&&C(D(1))<C(D(4))|
(C(D(1))==C(D(4))&&C(D(2))==C(D(3))&&C(D(2))<C(D(1))|
(C(D(1))==C(D(3))&&C(D(2))==C(D(4))&&C(D(1))<C(D(4))|
(C(D(3))==C(D(2))&&C(D(3))<C(D(4))&&C(D(4))<C(D(1))|
(C(D(1))==C(D(2))&&C(D(3))<C(D(1))&&C(D(1))<C(D(4))|
(C(D(4))==C(D(2))&&C(D(1))<C(D(3))&&C(D(3))<C(D(4))|
(C(D(1))==C(D(3))&&C(D(3))==C(D(4))&&C(D(2))<C(D(4))|
(C(D(4))==C(D(2))&&C(D(1))<C(D(2))&&C(D(2))<C(D(3))|
(C(D(4))==C(D(2))&&C(D(3))<C(D(1))&&C(D(1))<C(D(4))|
(C(D(4))==C(D(2))&&C(D(1))<C(D(2))&&C(D(2))<C(D(3))|
(C(D(4))==C(D(3))&&C(D(1))<C(D(2))&&C(D(2))<C(D(3))|
(C(D(4))==C(D(3))&&C(D(2))<C(D(1))&&C(D(1))<C(D(3))

```

z=z+2;

end

end

if length(find(C>0))==3

D=find(C>0);

```

if(C(D(1))<C(D(2))&&C(D(2))<C(D(3))|(C(D(2))<C(D(1))&&C(D(1))<C(D(3)))

```

z=z+2;

```

        end
        if
            (C(D(1))<C(D(3))&&C(D(3))<C(D(2)))|(C(D(2))<C(D(3))&&C(D(3))<C(D(1)))|(C(D(3))<C(D(1))&&C(D(1))<C(D(2)))|(C(D(1))<C(D(2))&&C(D(2))==C(D(3)))|(C(D(2))<C(D(1))&&C(D(1))==C(D(3)))|(C(D(3))<C(D(1))&&C(D(1))==C(D(2)))
                z=z+1;
        end
    end
    if length(find(C>0))==2
        D=find(C>0);
        if (C(D(1))<C(D(2)))
            z=z+1;
        end
    end
    z;
end
z
x1=(T(1,2)+T(1,3)+T(1,4)+T(1,5))/n;
x2=(T(2,3)+T(2,4)+T(2,5))/n;
x3=((T(3,4)+T(3,5)))/n;
x4=(T(4,5))/n;
if x1==round(x1)
    e1=e(1:n,1:x1);
end
if x1>round(x1)
    e1=e(1:n,1:(round(x1)+1));
end
if x1<round(x1)
    e1=e(1:n,1:round(x1));
end
[a1,aa1]=size(e1);
if x2==round(x2)
    e2=e(1:n,aa1+1:aa1+x2);
end

```

```

if x2>round(x2)
    e2=e(1:n,aa1+1:aa1+(round(x2)+1));
end
if x2<round(x2)
    e2=e(1:n,aa1+1:aa1+round(x2));
end
[a2,aa2]=size(e2);
if x3==round(x3)
    e3=e(1:n,aa1+aa2+1:aa1+aa2+x3);
end
if x3>round(x3)
    e3=e(1:n,aa1+aa2+1:aa1+aa2+(round(x3)+1));
end
if x3<round(x3)
    e3=e(1:n,aa1+aa2+1:aa1+aa2+round(x3));
end
if x4<round(x4)
    e3=e(1:n,aa1+aa2+1:aa1+aa2+round(x3));
end
[a3,aa3]=size(e3);
if x4==round(x4)
    e4=e(1:n,aa1+aa2+aa3+1:aa1+aa2+aa3+x4);
end
if x4>round(x4)
    e4=e(1:n,aa1+aa2+aa3+1:aa1+aa2+aa3+(round(x4)+1));
end
if x4<round(x4)
    e4=e(1:n,aa1+aa2+aa3+1:aa1+aa2+aa3+round(x4));
end
g1=e1%港口 1 堆箱矩阵
g2=e2
g3=e3
g4=e4%港口 4
[r1,u1]=size(g1);

```

```

[r2,u2]=size(g2);
[r3,u3]=size(g3);
[r4,u4]=size(g4);
% N=length(tem1);% 编码长度
%
N=(r1*u1+r2*u2+r3*u3+r4*u4)-sum(g1==0)-sum(g2==0)-sum(g3==0)-sum(g4==0);
%产生初始种群
Bestx=cell(1,5);
Besty=1e10;
Chrom=cell(pop,5);% 每列对应一个港口的取箱矩阵
for i=1:pop
    Chrom{i,1}=randperm(r1*u1);
    Chrom{i,2}=randperm(r2*u2);
    Chrom{i,3}=randperm(r3*u3);
    Chrom{i,4}=randperm(r4*u4);
    Chrom{i,5}=randperm(bay);
end
hwait=waitbar(0,'please wait.....');
steps=MaxGen;
%开始迭代求解
for k =1:MaxGen
    for s=1:pop

[val,A,Counttmove]=objection(Chrom{s,1},Chrom{s,2},Chrom{s,3},Chrom{s,4},Chrom{s,5});%求目标函数值
        J(s)=val;
    end
    Fit0=1./(J+1); % 求适应度值
    %轮盘赌选择
    totalfit =sum(Fit0);
    fitvalue =cumsum(Fit0);
    TempChrom=Chrom;
    for newin =1:pop
        fitin = 1;

```

```

temp0= rand*totalfit;
while temp0>fitvalue(fitin)
    fitin = fitin +1;
end
for jj=1:5
    TempChrom{newin,jj}=Chrom{fitin,jj};
end
end
%交叉
Pc = px;
for i =1:2:(pop-1)
    if Pc>rand
        for ii=1:4
            Temp11=TempChrom{i,ii};
            Temp12=TempChrom{i+1,ii};
            point=randi([2 length(Temp11)-1],1,2);%交叉点
            Min=min(point);
            Max=max(point);
            Temp0=Temp11(1:Min-1);
            Temp11(1:Min-1)=Temp12(1:Min-1);
            Temp12(1:Min-1)=Temp0;

            Temp0=Temp11(Max+1:end);
            Temp11(Max+1:end)=Temp12(Max+1:end);
            Temp12(Max+1:end)=Temp0;

            for j=1:Min-1
                while(find(Temp11(Min:Max)==Temp11(j)))
                    temp0=find(Temp11(Min:Max)==Temp11(j));
                    Temp11(j)= Temp12(temp0+Min-1);
                end

                while(find(Temp12(Min:Max)==Temp12(j)))
                    temp1=find(Temp12(Min:Max)==Temp12(j));

```

```

        Temp12(j)=Temp11(temp1+Min-1);
    end
end

for j1=Max+1:length(Temp11) %%%%
    while(find(Temp11(Min:Max)==Temp11(j1)))
        temp2=find(Temp11(Min:Max)==Temp11(j1));
        Temp11(j1)=Temp12(temp2+Min-1);
    end

    while(find(Temp12(Min:Max)==Temp12(j1)))
        temp3=find(Temp12(Min:Max)==Temp12(j1));
        Temp12(j1)= Temp11(temp3+Min-1);
    end
end

TempChrom{i,ii}=Temp11;
TempChrom{i+1,ii}=Temp12;
end
%bay 位分配
Temp11=TempChrom{i,5};
Temp12=TempChrom{i+1,5};
point=randi([2 bay-1],1,2);
Min=min(point);
Max=max(point);
Temp0=Temp11(1:Min-1);
Temp11(1:Min-1)=Temp12(1:Min-1);
Temp12(1:Min-1)=Temp0;

Temp0=Temp11(Max+1:end);
Temp11(Max+1:end)=Temp12(Max+1:end);
Temp12(Max+1:end)=Temp0;

for j=1:Min-1
    while(find(Temp11(Min:Max)==Temp11(j)))

```



```

        temp0=find(Temp11(Min:Max)==Temp11(j));
        Temp11(j)= Temp12(temp0+Min-1);
    end

    while(find(Temp12(Min:Max)==Temp12(j)))
        temp1=find(Temp12(Min:Max)==Temp12(j));
        Temp12(j)=Temp11(temp1+Min-1);
    end
end

for j1=Max+1:bay %%%%
    while(find(Temp11(Min:Max)==Temp11(j1)))
        temp2=find(Temp11(Min:Max)==Temp11(j1));
        Temp11(j1)=Temp12(temp2+Min-1);
    end

    while(find(Temp12(Min:Max)==Temp12(j1)))
        temp3=find(Temp12(Min:Max)==Temp12(j1));
        Temp12(j1)= Temp11(temp3+Min-1);
    end
end
TempChrom{i,5}=Temp11;
TempChrom{i+1,5}=Temp12;
end
end
% 变异
Pm = pm;
for i =1:pop
    if Pm>rand
        for j=1:5
            tem=TempChrom{i,j};
            point=randi([2,length(tem)-1],1,2);
            p1=max(point);
            p2=min(point);

```

```

        TempChrom{i,j}=[tem(p2+1:p1) tem(1:p2) tem(p1+1:end)];
    end

    %
    TempChrom(i,1+(G-1)*(G-1):end)=fliplr(TempChrom(i,1+(G-1)*(G-1):end));
    end
end

[value,ind]=min(J);

[Temp_val,A,Counttmove]=objection(TempChrom{ind,1},TempChrom{ind,2},TempChrom{ind,3},TempChrom{ind,4},TempChrom{ind,5});

if value<Temp_val
    for kk=1:5
        TempChrom{ind,kk}=Chrom{ind,kk};
    end
end

if value<Besty
    Besty=value;
    for kk=1:5
        Bestx{1,kk}=Chrom{ind,kk};
    end
end

Chrom=TempChrom;
Jlist(k)=Besty;
str=[num2str(round((k/steps)*100)),'%'];
waitbar(k/steps,hwait,str);
end

close(hwait);
% 结果
disp('最优取箱矩阵: ')
Result1=[g1(Bestx{1,1}) g2(Bestx{1,2}) g3(Bestx{1,3}) g4(Bestx{1,4})]
disp('最优装箱顺序（矩阵元素由 bay 位编号组成）: ')
ord=Bestx{1,5}
[result_val,rrc,Counttmove]=objection(Bestx{1,1},Bestx{1,2},Bestx{1,3},Bestx{1,4},Bestx{1,5});
disp(['最优目标函数: ',num2str(result_val)])

```

```
disp(['岸桥贝位间来回移动次数:',num2str(Counttmove)])  
disp('装载矩阵')  
rrc  
figure(1)  
plot(1:length(Jlist),Jlist,'r','linewidth',2)  
xlabel('迭代次数');  
ylabel('每代最优目标函数值');  
title('遗传算法收敛图')
```