

路径优化算法求解集装箱码头堆场翻箱问题

郑斯斯, 王爱虎

(华南理工大学 工商管理学院, 广东 广州 510640)

摘要: 集装箱翻箱问题(CRP)可描述为在集装箱堆场现有堆垛状态和提箱序列确定情况下,以最少的翻箱数量提取出堆场箱区内所有集装箱;CRP 是一个 NP-hard 问题。为此构建双层目标规划模型,提出嵌套翻箱规则的路径规划算法(POA),以期减少解空间大小,从而在更短的 CPU 运行时间内得到 CRP 的最优解。数值实验结果表明,POA 在翻箱数量及运行时间上均优于多数算法,有效提高集装箱码头堆场作业效率,更适用于求解集装箱码头翻箱作业优化问题。

关键词: 翻箱问题; 提箱序列; 启发式规则; 路径优化算法

中图分类号: U656.1+35

文献标识码: A

Paths Optimum Algorithm for the Container Relocation Problem in the Container Terminals

ZHENG Si-si, WANG Ai-hu

(School of Business Administration, South China University of Technology, Guangzhou 510640, China)

Abstract: The container relocation problem (CRP) can be defined as a sequence which allows each container to be extracted with least number of relocations when a cluster of identical containers have been stacked in a storage area. CRP is an NP-hard problem. Firstly, a two-level goal programming model for CRP is presented that can help understand this problem. Secondly, on account of the complexity of CRP, heuristic rules are proposed to reduce hunting space by the way of dividing solution space. Thirdly, paths optimum algorithm based upon these heuristics is proposed to reduce unfeasible solutions and solve optimal solution for any instances in a shorter CPU runtime. Finally, numerical tests are used to compare results of POA and other algorithms, showing that POA is better in the areas of efficiency, robustness and accuracy. Furthermore, POA improves the operation efficiency of container terminals effectively, and it is suitable for container pick-up optimization.

Key words: container relocation problem; extractive sequence; heuristic rules; paths optimum algorithm

1 引言

随着 2002 年以来港口管理属地化,我国港口实

现了跨越式发展。就珠三角而言,随着深圳、珠海等海港城市的兴起,珠三角港口群整体已呈现出港口能力过剩隐患,特别是随着大规模实施新港区开发

收稿日期:2016-09-29; 修回日期:2016-11-19

基金项目:国家自然科学基金资助项目(71171084);广州市金融服务创新与风险管理研究基地 2016 年度研究课题(2016GZJFJD02)

作者简介:郑斯斯(1990-),广东汕头人,博士研究生,主要研究方向为物流与供应链管理和最优化理论, E-mail: zheng.sisi@mail.scut.edu.cn。

王爱虎(联系人),教授, E-mail: bmawang@scut.edu.cn。

及港口相关产转移升级,造成港口数量快速膨胀,结构性过剩突出,竞争日趋激烈。在全球经济复苏缓慢大背景下,集装箱码头内部效率的提升成为港口经营者获取竞争优势的重要手段。为提升集装箱码头自身竞争力,港口需要使用最低成本支出获取最高效率收益,而堆场管理问题直接影响集装箱运输成本和效率。

由于船舶抵港时间具有不确定性,可能出现集装箱提箱顺序与初始堆放位置不完全相符的情况,导致需要较早离场集装箱被压在后续离场集装箱下层现象。所以提箱过程中不可避免需要进行翻箱,甚至当被翻箱落箱位置选取不当,会造成后续提箱的二次翻箱。因此为提高集装箱装船效率的同时减少翻箱操作,通常会充分利用船舶进港前空闲时间对集装箱进行翻箱,以保证整理后堆场状态与装船顺序一致,这一类问题称为翻箱问题。以某集装箱码头为例,假设码头年吞吐量为 30 万 TEU,平均每次翻箱耗时约为 4 分钟,翻箱成本约为 20 元,若每年翻箱率降低 2%,可节约翻箱成本高达 12 万元,节约额外翻箱时间长达 400 小时。因此如何最大限度地减少翻箱量,从而减少作业时间,以实现提高整体效率,进而降低营运成本,成为港口管理者最大的目标。

在对珠三角多个集装箱码头进行现场调研基础上,本文以集装箱码头龙门吊堆场为研究对象,围绕堆场提箱及翻箱率问题,探讨提箱过程建模和优化方法,为沿海和内河集装箱码头运作效率的提升提供方法论。

2 翻箱问题研究现状

翻箱问题是港口运营必须面临的挑战,科学分析翻箱移动路径、优化翻箱作业有助于码头管理者有效提高堆场整体效率,使企业在竞争当中处于优势。近年来学术界对集装箱码头堆场翻箱问题的研究日趋增多。

2.1 国际研究现状

国际学术界主要通过检索集装箱堆垛位置以实现集装箱翻箱优化。部分文献利用精确算法,求得全局最优解。Kim 等^[1-2]在 1997 年提出估计翻箱量的方法,通过拉格朗日松弛法求得栈高,以次梯度法优化栈高并梳理栈高与翻箱量之间的关系;考虑到装船过程中重箱不压轻箱, Kim 等^[3]建立以最小化装船过程翻箱量为目标的数学模型,通过动态规划法优选集装箱箱位;部分文献采用近似算法,

Caserta 等^[4-6]针对翻箱问题提出基于通道的启发式算法; Lee and Hsu^[7]建立以最小化预翻箱过程翻箱量为目标的整数规划模型,开发启发式算法优化翻箱路径; Yang and Kim^[8]针对提箱过程中被翻箱的落箱位选择,提出 OH 算法对落箱位置进行优选; Lee and Chao^[9]采用领域搜索启发式算法优化预翻箱问题; Carlo 等^[10]对 2004~2012 年间公开发表的文献进行研究,汇总并归纳所提出的解决方法、模型、算法等,基于当前行业发展趋势和集装箱码头研究情况确定学术研究途径。

2.2 中国研究现状

国内学术界侧重于分析实际作业情况并对实操结果进行总结,研究多以近似算法为主。王新颖等^[11]分析了集装箱到达堆场的概率分布,对翻箱率进行仿真,并为箱位分配提出建议;林燕等^[12]针对集装箱装卸作业调度问题,提出了集成优化调度的启发式算法;沙梅等^[13]将优化方法与仿真技术方法相结合,对集装箱码头堆场的布局进行了优化和仿真研究;边展等^[14]针对多台龙门吊同时作业以提升装船效率的情况进行深入分析,采用了两阶段动态规划算法求解取箱调度问题;邵乾虔等^[15]在不改变集装箱初始堆垛状态的同时考虑海关因素,建立与最小化翻箱量为目标的优化模型,用启发式算法进行求解;李浩渊等^[16]引入“混乱系数”的概念作为目标函数,对预倒箱问题进行约束,构建了数学规划模型,开发二元编码方式的多阶段遗传算法进行求解。

2.3 评价方法研究现状

国内外学者关于集装箱堆场的翻箱问题主要采用的优化方法有以下两类:①精确算法,如动态规划算法、分支定界算法等,这类方法最大优势在于针对小规模翻箱问题能求得全局最优解,从而得到精确的翻箱数量;在堆场规模增大时,算法复杂度增大,该类研究方法出现瓶颈;②近似算法,如模拟退火算法、仿真等,这类方法主要是根据翻箱问题计算复杂度大的特征,采用智能算法近似求得最优解;薄弱之处在于其表现不稳定,其可能出现针对同一状态有不同结果。综合分析国内外研究情况,目前学者们对翻箱问题已经做过相应实践研究,不同方法在不同程度上都能提高翻箱效率。研究仍存在不足之处,由于不同集装箱码头实际情形不尽相同,多数研究成果并没有考虑集装箱不同属性,导致实际操作中无法将研究成果进行实践。为此本文结合堆场实

际情形,采用启发式算法研究集装箱翻箱问题。

3 翻箱问题描述

在集装箱码头堆场中,垂直堆垛的一列集装箱称一个栈;并排堆放的若干个栈称一个贝;贝中水平堆垛的集装箱被称为一层,港口通常采用每个栈含3~6层的多层集装箱堆垛方式,如图1(a)所示。码头运营者根据集装箱尺寸、重量等不同信息进行区分,将

相同属性集装箱尽可能集中堆垛,以提高作业效率。最常用的装卸设备是龙门吊,龙门吊包括大车和小车,其中小车在垂直方向上行走,即在同贝不同栈之间进行作业;大车在水平方向上行走,即在不同贝之间进行作业,龙门吊行走距离包括水平和垂直方向,如图1(b)所示。集装箱离港时,集装箱由龙门吊从堆场吊起并放到集卡,再运送到船上。根据堆场实际情形,翻箱问题理论基础依赖于以下前提。

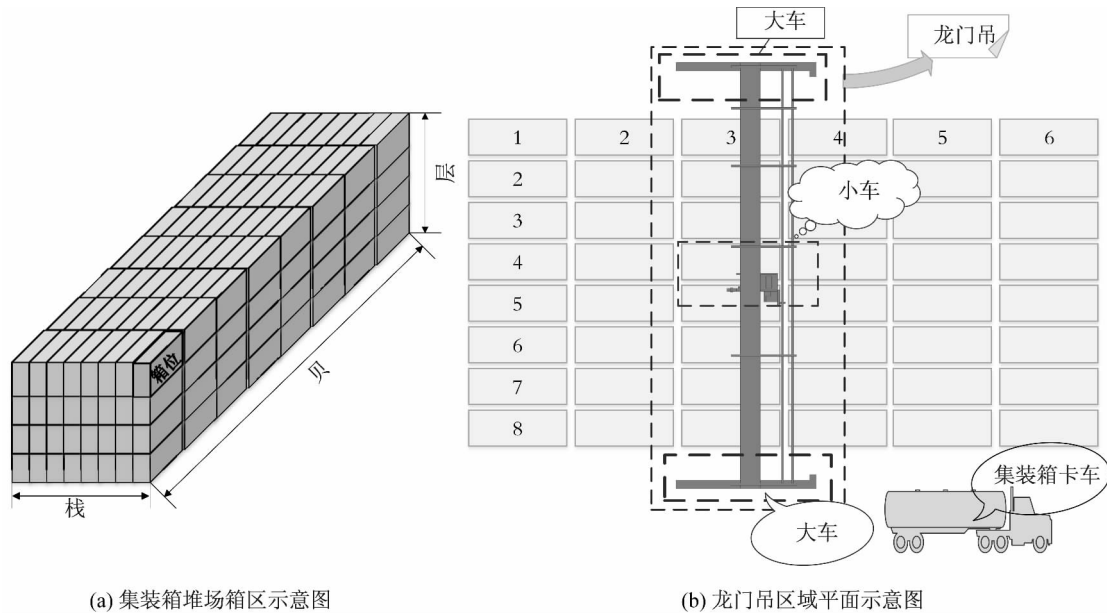


图1

- (1) 箱区内贝、栈、层数量已定。
- (2) 每一箱位只能堆垛一个20尺集装箱。
- (3) 由于龙门吊大车走动速度慢,翻箱尽量在同贝内进行。
- (4) 每一栈只能堆垛到给定最高层数,当达到堆垛上界,集装箱只能堆到其余栈。
- (5) 翻箱作业只发生在最顶层。
- (6) 箱区内至少预留 $(K-1)$ 个空箱位便于翻箱使用。
- (7) 提箱前,集装箱在箱区内分布及后续提箱顺序已知。
- (8) 提箱时,箱区内没有新箱子进入堆场。

4 模型相关定义

4.1 基本概念定义

定义1 箱位矩阵。设箱区内有 m 贝 n 栈 k 层,则堆存状态可用矩阵 H 表示。 H 中每个元素代表该箱位上集装箱优先级别。 $H(m, n, k)=0$ 表示箱位上没有集装箱, $H(m, n, k)=h > 0$ 表示箱

位堆放优先级为 h 的集装箱。则有:

$$H = \{g_n | g_n \in \mathfrak{S}\}, g_n = \{\tau_n | \tau_n \in \partial\}, \\ \tau_n = \{g_{kn} | g_{kn} \in \Phi\}$$

其中 \mathfrak{S} 表示贝, ∂ 表示栈, Φ 表示层。

定义2 优先级。给定矩阵 H ,以数字 $1, 2, \dots, S$ 代表所有集装箱提箱顺序,且每个集装箱优先级唯一确定。

定义3 正常序。同一箱区集装箱满足按优先级从高到低进行堆垛的顺序。

定义4 目标箱。某一堆存状态下,应最先提取的集装箱称该堆存状态下目标箱。

定义5 阻塞箱。同一栈中,若集装箱下方存在比其需更早取走箱子,则该集装箱称该堆存状态下阻塞箱,又称被翻箱。

4.2 基本符号定义

为方便描述,如表1所示定义了基本符号,如表2所示定义了决策变量,其中 $m=0, 1, 2, \dots, M, n=0, 1, 2, \dots, N, k=0, 1, 2, \dots, K, z=0, 1, 2, \dots, k-1$ 。

4.3 双层目标规划模型

翻箱问题可分为两层:具体贝位分配和具体箱位分配。鉴于两层问题存在关联性,因此构建双层

表 1 定义翻箱问题基本符号

基本符号	符号说明
S	箱区内集装箱数量($s=1,2,\dots,S$ 且 $N \times (K-1) \geq S$)
M	箱区内贝位数量($m=1,2,\dots,M$)
N	单贝内栈位数量($n=1,2,\dots,N$)
K	单栈层数($k=1,2,\dots,K$)
C	一个充分大正数, $C > 0$
$P(s)$	第 s 个集装箱优先级($P(s)=1,2,\dots,S$)
$V(m)$	第 m 贝最大容量
$slot(m,n,k)$	第 m 贝 n 栈 k 层的箱位

表 2 定义翻箱问题决策变量

决策变量	符号说明
S_{smnk}	表示 $slot(m,n,k)$ 是否被第 s 个集装箱占据。 $S_{smnk} = \begin{cases} 1 & \text{当第 } s \text{ 个集装箱分配到箱位 } slot(m,n,k) \\ 0 & \text{当第 } s \text{ 个集装箱分配到其它箱位} \end{cases}$
$R_{nm(k-z)}$	表示堆放在箱位 $slot(m,n,k)$ 与 $slot(m,n,k-z)$ 上的集装箱之间是否存在翻箱。假设集装箱 a 堆放在 $slot(m,n,k)$, 集装箱 b 堆放在 $slot(m,n,k-z)$ 。 $R_{nm(k-z)} = \begin{cases} 1 & P(a) < P(b) \\ 0 & \text{其它} \end{cases}$

数学规划模型,优化目标如表 3 所示。

表 3 双层规划模型的优化目标

不同层	优化目标
第一层	优化箱区内总翻箱数量及龙门吊运作距离
第二层	优化贝内的翻箱数量

首先讨论第一层,由于小车运行速度约为大车的 2—3 倍,同等距离下大车比小车更耗时。令 α, β 表示不同贝、不同栈之间距离权重系数且 $\alpha > 4\beta$, 则有:

$$X_s = \sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^K m \cdot (S_{smnk} - S_{(s-1)nmk})$$

$$Y_s = \sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^K n \cdot (S_{smnk} - S_{(s-1)nmk})$$

$\sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^K m \cdot S_{smnk}$ 和 $\sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^K n \cdot S_{smnk}$ 表示第 s 个集装箱堆垛在 m 贝 n 栈,此时龙门吊大车停留在 m 贝,小车停留在 n 栈; X_s 和 Y_s 表示两个相邻优先级别的集装箱移动的贝位差及栈位差,是两次相邻堆垛作业之间龙门吊大车、小车的移动距离。则目标函数为

$$Z_1 = \min \sum_{s=1}^S [\sqrt{(\alpha X_s)^2 + (\beta Y_s)^2}]$$

$$Z_2 = \min \sum_{m=1}^M \sum_{n=1}^N \sum_{k=2}^K \sum_{z=1}^{k-1} R_{nm(k-z)}$$

Z_1 表示最小化龙门吊移动距离,并赋予大车和小车权重系数, Z_2 表示最小化集装箱堆场箱区总翻箱数量。为保证每一次作业中提取集装箱的合理性,提出第一层约束条件。

(1) 任一集装箱 s 只能堆在一个箱位中:

$$\sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^K S_{smnk} = 1 \quad (1)$$

(2) 任一箱位最多只能堆一个集装箱:

$$\sum_{s=1}^S S_{smnk} \leq 1 \quad (2)$$

(3) 龙门吊大车初始位置在第 1 贝:

$$\sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^K m \cdot S_{1nmk} = 1 \quad (3)$$

然后讨论第二层,目标函数和约束条件为

$$Z_3 = \min \sum_{n=1}^N \sum_{k=2}^K \sum_{z=1}^{k-1} R_{nm(k-z)}$$

(4) 每贝集装箱数量不超过该贝额定容量:

$$\sum_{n=1}^N \sum_{k=1}^K \sum_{s=1}^S S_{smnk} \leq V_m \quad (4)$$

(5) 任一集装箱不允许悬空放置:

$$\sum_{s=1}^S S_{smnk} \leq \sum_{s=1}^S S_{smn(k-1)} \quad (5)$$

(6) 集装箱 a 和 b 堆垛在 $slot(m,n,k)$ 和 $slot(m,n,k-z)$, 且 $P(a) < P(b)$, 则移走 a 时,需先移走 b ,记一次翻箱 $R_{nm(k-z)} = 1$:

$$\sum_{s=1}^S P(s) \cdot (S_{smnk} - S_{smn(k-z)}) \geq (\sum_{s=1}^S S_{smnk} - 1 - R_{nm(k-z)}) \cdot C \quad (6)$$

综合考虑,双层数学规划模型可表述为

$$\begin{cases} Z_1 = \min \sum_{s=1}^S [\sqrt{(\alpha X_s)^2 + (\beta Y_s)^2}] \\ Z_2 = \min \sum_{m=1}^M \sum_{n=1}^N \sum_{k=2}^K \sum_{z=1}^{k-1} R_{nm(k-z)} \\ Z_3 = \min \sum_{n=1}^N \sum_{k=2}^K \sum_{z=1}^{k-1} R_{nm(k-z)} \\ \text{s. t. (1) - (6)} \end{cases}$$

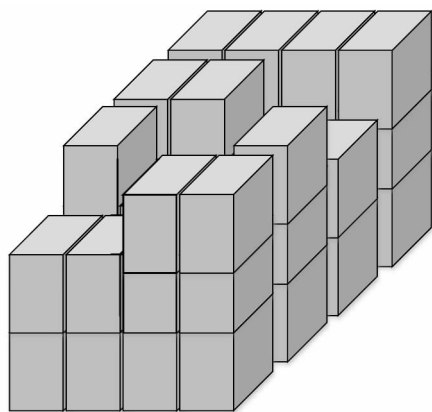
5 启发式规则

Caserta 等^[5]一文已证明 CRP 是一个 NP-hard 问题,而翻箱路径优化问题是其扩展,难以在多项式时间内求得最优解。因此作者结合现场调研经验,提炼翻箱规则并设计启发式算法以减少提取路径。以下的翻箱规则将有助于选取最优堆垛箱位。

5.1 翻箱规则

(1) 同贝优先, 邻贝次之原则 (Same Bay First & Neighboring Bays Second, SFNS)。该规则下移动阻塞箱, 当同贝内不同栈仍有空箱位, 优先翻到本贝; 当同贝不同栈没有空箱位, 则再考虑翻到相邻贝。由于龙门吊大车的走动更耗时, 对以效率为先的集装箱码头而言, 节省时间是提高效率的方式之一。

(2) 阻塞箱较少原则 (Least - Obstructive Heuristic, LOH)。对周转率极高的集装箱堆场而言, 时间就是效率, 因此提取目标箱时, 如果本栈内阻塞箱越少, 造成的翻箱量越小, 耗时就越少。当阻塞箱在同贝内寻找落箱位位置时, 根据集装箱优先级, 比较被翻箱与同贝内不同栈集装箱优先级别。当被翻箱落到贝内第一栈, 若被翻箱优先级只低于该栈一个集装箱, 记 $LO(1)=1$, 比两个箱低, 记 $LO(1)=2$, 以此类推, 记录被翻箱落到第一栈阻塞箱数量 $LO(1)$; 直至记录被翻箱落到贝内最后一栈阻塞箱数量 $LO(n)$ 。寻最小 $LO(i)$, 栈 i 为 LOH 下被翻箱落箱最优栈。其中以下两种情况记 $LO(x)=inf$: ① 被翻箱仍停在原箱位; ② 某栈已达到最高层。注意到, 若没有信息系统和智能算法支持, 操作手不会考虑到该规则, 更多凭自身经验, 可能导致翻箱量增多, 因此 LOH 具有重大意义。



(a) 箱区的一种堆存状态

(3) 就近堆垛原则 (Nearest - Slot Heuristic, NSH)。为操作便捷, 操作手一般将阻塞箱堆到同贝内离原箱位最近栈的空箱位上。计算被翻箱落箱位置所在栈与原箱位所在栈之差, 历遍每一栈并寻求最小差 $NS(i)$ 。在高速周转集装箱码头, NSH 能减少龙门吊小车单次作业移动距离, 则作业时间越短。

(4) 最矮栈堆垛原则 (Lowest - Slot Heuristic, LSH)。通常在集装箱进场前, 进场、逗留、离场时间等信息不多, 出于安全和方便考虑, 操作手会将集装箱堆在同贝内最矮栈, 保证同贝内集装箱平均高度最低并减少龙门吊提升高度。计算被翻箱落箱位置所在高度, 历遍每一栈并寻求最小差 $LS(i)$ 。堆场的平均堆垛高度越低, 发生事故概率越低。

引入科学有效启发式方法^[17-18]优选被翻箱落箱位置, 以减少无效路径, 从而降低计算复杂度。这些翻箱规则都具有重要现实意义, 因此何时选择何种规则将会影响提箱效率。

5.2 翻箱示例

如图2所示为4贝4栈3层箱区的初始堆存状态, 集装箱分别从1—42号给定了优先级, 0表示该箱位为空箱位。优化目标是将全部集装箱按顺序提走且翻箱数量最小。

0	0	14	25	2	0	0	8
1	4	17	36	34	39	40	29
33	6	20	11	27	15	9	23
Bay 1				Bay 2			
21	7	0	0	12	41	35	26
42	24	32	37	38	3	28	13
5	22	18	19	16	31	30	10
Bay 3				Bay 4			

(b) 该箱区内的集装箱优先级状态

图2 4贝4栈3层箱区的初始堆存状态

针对该例, 图3详细描述如何提走集装箱。

6 路径优化算法

近年来的研究中, 越来越多学者采用启发式算法求解 CRP, 如 Caserta 等^[19], Wang 等^[20], 祝慧灵等^[21], 范厚明等^[22]。

翻箱问题研究难度在于尽管同贝内集装箱初始状态已定, 仍有大量的可行提取路径, 导致计算很耗时; 且 NP-hard 问题很难求解, 使得翻箱问题很难找到多项式时间算法。在本节作者提出路径优化算法以实现图3所示的操作, 流程如图4所示, 程序伪代码如表4所示。



图 5 是以翻箱规则顺序(1)为例的算法流程, 输入一组同贝初始状态矩阵, 输出翻箱数量及运行时间。同理分别输入余下五种顺序, 比较同贝内同组初始状态下翻箱数量及运行时间, 得到最优顺序。

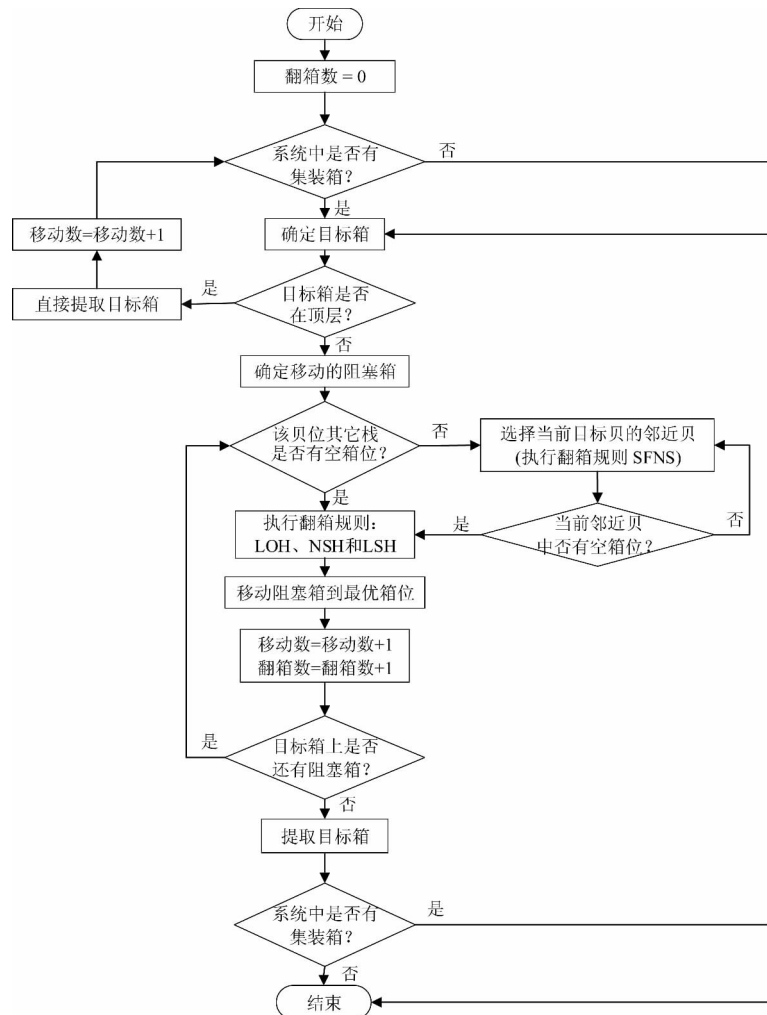


图4 嵌套翻箱规则的提箱流程

7 算例分析

7.1 比较不同的翻箱规则顺序

首先比较同贝内所有翻箱规则顺序。设贝内有 N 栈 K 层, 集装箱 S 个分别堆在该贝内。生成 $N \times K - (K-1)$ 个集装箱, 且每个集装箱的离场顺序已定, 有唯一的优先级别。随机生成 100 组实例, 执行 POA 并记录对应的翻箱数量。采用同组例子测试六种翻箱规则顺序性能, 从而找到最优翻箱规则顺序, 数据如图 6 所示。

翻箱规则优先级别为 (1) → (2) → (6) → (5) → (4) → (3)。因此采用 SFNS → LOH → LSH → NSH 顺序能得到最少翻箱数量, 此时 POA 性能达到最优。

从表中可见 POA 数据几乎都超越其它算法, 占据绝对优势。一方面, 由于翻箱操作成本高, 所以 POA 能节约成本; 另一方面, POA 能在短时间内解决 CRP, 若引用到集装箱码头, 操作手能在更短时

间内决定被翻箱落箱位置。

7.2 比较 POA 与已有算法

现有研究多个算法只研究同贝内翻箱作业, 因此我们在单贝情况下比较 POA 与多个算法性能。KH、DH、CM、LA 算法分别由 Kim and Hong^[21]、Aydin and Ünlüyurt^[23]、Caserta 等^[19]、Matthew 等^[24]提出。数值实验采用 100 个随机生成状态矩阵, 测试结果如表 5 所示。

7.3 灵敏度分析

由于翻箱数量与箱区规模关系密切, 为考察 POA 整体性能, 通过改变集装箱堆场的利用率, 以比较不同箱区规模的翻箱量和运行时间。测试结果如表 6 所示。

POA 不仅能控制翻箱数量, 且将运行时间保持在稳定低水平, 在堆存状态为 21 贝 10 栈 8 层, 箱区容量达 1680 TEU, 运行时间在 13 秒以内。

图 7 为不同堆场利用率与翻箱数量关系。

表 4 路径优化算法程序的伪代码

算法:针对翻箱问题的路径优化算法

输入:箱位矩阵(任一状态)、集装箱总数量。

变量: H 表示三维矩阵; $H(bay, stack, tier)$ 表示集装箱箱位; S 表示总集装箱数量; $xthcontainer$ 表示集装箱优先级,即第 x 号集装箱; $N(m)$ 表示集装箱移动总次数; $N(r)$ 表示集装箱翻箱总次数; $F(g) = \begin{cases} 1 & \text{直接提取目标箱} \\ 0 & \text{其它} \end{cases}$

输出:集装箱移动总次数;集装箱翻箱总次数;最优翻箱路径;运行时间

1:开始计时

2:初始设置 $N(m) \leftarrow 0, N(r) \leftarrow 0, xthcontainer \leftarrow 1$ 3:for 集装箱数量 $\leq Sdo$ 4:寻找目标箱并记录 $H(bay, stack, tier)$ 和 $xthcontainer$ 5:if $F(g) = 1$ then6: $H(bay, stack, tier) := 0$ 7: $N(m) \leftarrow N(m) + 1$ 8:记录 $H(bay, stack, tier)$ 和 $xthcontainer$ 并更新矩阵 H

9:else

10:根据上文“提取集装箱”的流程,移动阻塞箱(移动阻塞箱并记录最优翻箱路径,移动 $(athbay, athstack, athtier)$ 到 $(bthbay, bthstack, bthtier)$)11: $N(m) \leftarrow N(m) + 1$ 12: $N(r) \leftarrow N(r) + 1$ 13:记录 $H(bay, stack, tier)$ 和 $xthcontainer$ 并更新矩阵 H

14:end if

15:end for

16:直到系统内没有集装箱

17:返回 CRP 最优解

18:结束计时

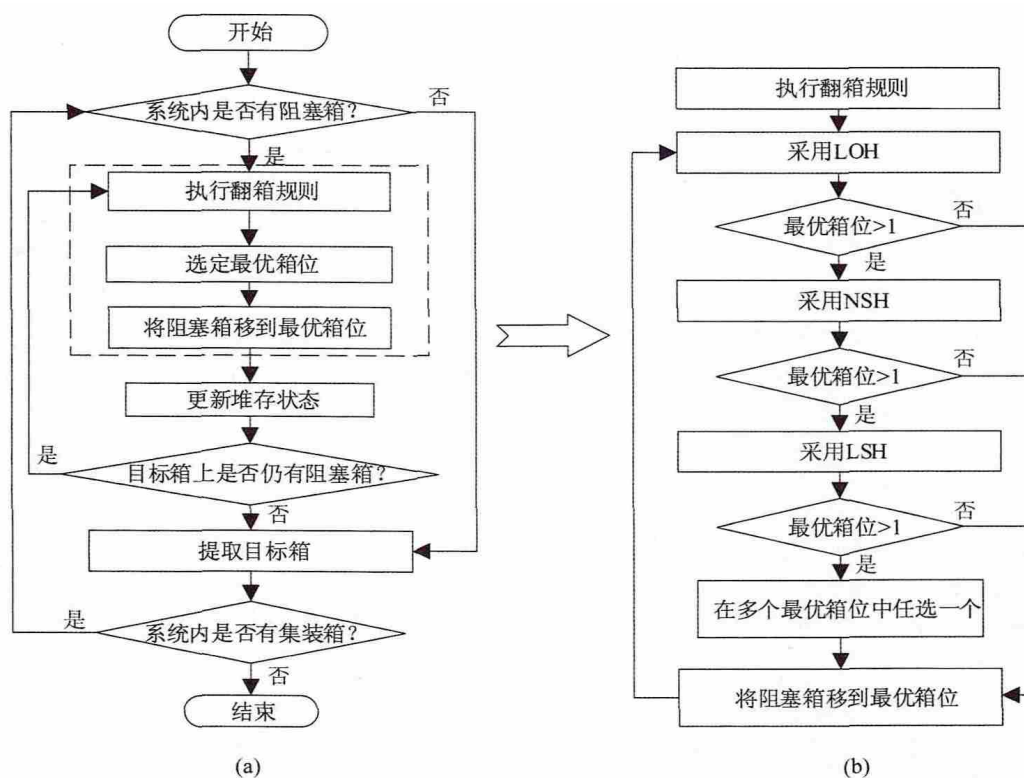


图 5 以翻箱规则顺序(1)为例的算法流程

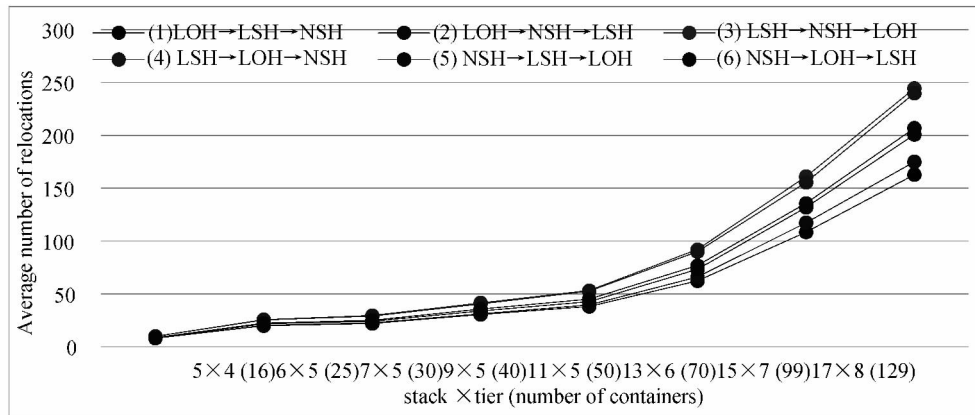


图6 贝内不同规模下集装箱平均翻箱数量

表5 堆场同贝任一实例下平均翻箱数量及运行时间

贝位规模 $stack \times tier$	KH		DH		CM		LA		POA	
	No.	Time	No.	Time	No.	Time	No.	Time	No.	Time
3×3	7.1	0.1	5.6	<1	5.1	0.1	5.4	<1	3.33	0.029
4×3	10.7	0.1	7.3	<1	6.3	0.1	6.5	<1	4.125	0.027
5×3	14.5	0.1	8	<1	7	0.1	7.3	<1	5.75	0.035
4×4	26	0.1	12.8	<1	10.6	0.2	10.5	<1	10.4	0.038
5×4	16	0.1	12.2	<1	10.4	0.2	9.9	<1	8.049	0.032
6×4	23.4	0.1	15.7	<1	13	0.5	16.5	<1	8.714	0.037
6×5	37.5	0.1	23.9	<1	18.8	0.8	19.7	<1	15.2	0.037
7×5	45.5	0.1	27.9	<1	22.1	0.8	22.6	<1	19.83	0.039
8×5	52.3	0.1	31.9	<1	25.8	1.43	24.8	<1	21.88	0.044
10×6	80.9	0.1	45.2	<1	36.4	1.87	33.5	<1	30.6	0.060
10×10	75.1	0.1	61.5	1.95	49.5	1.95	46.8	<1	44.8	0.073

No. : 100 个实例的平均翻箱数量 (TEU); Time: 电脑测试运行时间 (秒)

表6 堆场多贝平均翻箱数量及运行时间

堆场利用率	50%		60%		70%		80%		90%	
箱区规模	No.	Time	No.	Time	No.	Time	No.	Time	No.	Time
13×6×4	46	0.32	75.8	0.42	94	0.48	138.29	0.58	187.29	0.68
15×7×5	103.9	0.85	144.3	0.98	222.1	1.19	289.57	1.39	384.14	1.60
17×8×6	195.7	1.54	296.7	1.88	400.2	2.33	545.86	2.69	723.14	3.13
19×9×7	332.6	3.11	495.9	3.97	694.1	4.61	925.43	5.73	1221.4	6.78
21×10×8	532.9	6.18	790	7.54	1095	9.48	1526.9	11.1	1922.8	12.5

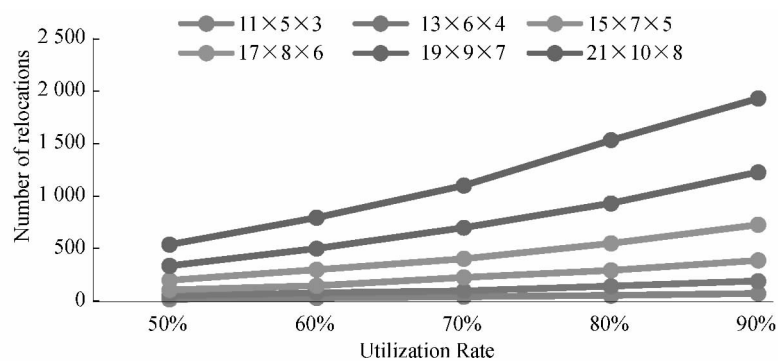


图7 不同堆场利用率与集装箱翻箱数量关系

随着箱区规模的扩大,所有的曲线都加快上升的速度,尤其当利用水平达到80%,曲线飞速上升。

曲线越倾斜,翻箱数量更大,所以堆场利用率不宜过大。将堆场箱区的利用率控制在 70%~80%之间,堆场的效率达到最优。

8 结论

翻箱作业是影响集装箱码头龙门吊堆场作业效率的重要因素,目前国内外学术界对该问题已经有很多实践研究,但由于不同码头的实际情形不尽相同,多数研究成果无法直接应用在集装箱码头。为此,本文结合集装箱码头作业实际状况,构建双层目标规划模型并提出了嵌套翻箱规则的路径优化算法进行求解,以期减少堆场翻箱数量,进而提高码头作业效率。数值结果表明模型不仅能有效降低该类问题的解空间大小,缩短求解问题的时间复杂度,在翻箱数量及 CPU 运行时间上优于很多现有算法,因此,POA 是一个解决 CRP 的更优算法。POA 还能作为集装箱码头翻箱作业优化调度问题提供决策支持,有效实现理论与实际操作相结合。

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, Kim H B. Segregating space allocation models for container inventories in port container terminals [J]. International Journal of Production Economics, 1999, 59(1-3): 415-423.
- [3] 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.
- [4] Caserta M, Schwarze S, Voß S. A new binary description of the blocks relocation problem and benefits in a look ahead heuristic [C]//Proceedings of the 9th European Conference on Evolutionary Computation in combinatorial Optimization. Berlin, Germany: Springer-Verlag, 2009, 5482: 37-48.
- [5] Caserta M, Schwarze S, Sniedovich M. Applying the corridor method to a blocks relocation problem[J]. OR Spectrum, 2011, 33(4): 915-929.
- [6] Caserta M, Voß S. A Corridor Method-Based Algorithm for the Pre-marshalling Problem [J]. Applications of Evolutionary Computing, 2009, 5484: 788-797.
- [7] Lee Y S, Hsu N Y. An optimization model for the container pre-marshalling problem [J]. Computers & Operations Research, 2007, 34(11): 3295-3313.
- [8] Yang J H, Kim K H. A grouped storage method for minimizing relocation in block stacking system[J]. Journal of Intelligent Manufacturing, 2006, 17(4): 453-463.
- [9] Lee Y S, Chao S L. A neighborhood search heuristic for pre-marshalling export containers [J]. European Journal of Operational Research, 2009, 196(2): 468-475.
- [10] Carlo H J, Vis I F A, Roodbergen K J. Storage yard operations in container terminals: Literature overview, trends, and research directions [J]. European Journal of Operational Research, 2014, 235(2): 412-430.
- [11] 王新颖. 集装箱场站翻箱率仿真研究[D]. 大连: 大连海事大学, 2008.
- [12] 林燕, 孙小明. 新型集装箱码头堆场混合装卸计划方法研究[J]. 工业工程与管理, 2011, 16(5): 97-101.
- [13] 沙梅, 周鑫, 秦天保. 集装箱码头堆场布局优化与仿真研究[J]. 工业工程与管理, 2013, 18(2): 24-30.
- [14] 边展, 杨惠云, 靳志宏. 基于两阶段混合动态规划算法的龙门吊路径优化[J]. 运筹与管理, 2014, 23(3): 56-63.
- [15] 邵乾虔, 徐奇, 边展, 等. 考虑了交箱时间不确定性的场桥堆存作业优化[J]. 系统工程理论与实践, 2015, 35(2): 394-405.
- [16] 李浩渊, 王洪峰, 孙冬石. 集装箱预倒箱问题的多阶段遗传算法研究[J]. 控制工程, 2015, 22(4): 700-704.
- [17] 侯东亮, 邹律龙. 求解集装箱堆场堆垛问题的约束满足算法[J]. 工业工程与管理, 2012, 17(4): 84-88.
- [18] 张小杰. 港口机械智能维护的决策支持方法[J]. 工业工程与管理, 2012, 17(5): 136-140.
- [19] Caserta M, Voß S, Sniedovich M. Applying the corridor method to a blocks relocation problem[J]. OR Spectrum, 2011, 33(4): 915-929.
- [20] Wang N, Jin B, Lim A. Target-guided algorithms for the container pre-marshalling problem[J]. Omega, 2015, 53: 67-77.
- [21] 祝慧灵, 计明军, 郭文文, 等. 基于配载计划的集装箱提箱顺序和倒箱策略优化[J]. 交通运输系统工程与信息, 2016, 16(2): 191-199.
- [22] 范厚明, 姚茜, 马梦知. 多场桥分区域平衡策划下的集装箱堆场箱位分配问题[J]. 控制与决策, 2016, 31(9): 1603-1608.
- [23] Aydin C, Ünlüyurt T. Improved Rehandling Strategies for Container Retrieval Process [C]. Turkey: Sabanci University, 2008.
- [24] Matthew E H, Mazon I H. A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem [J]. European Journal of Operational Research, 2013, 231(1): 120-130.