超星·期刊

Vol. 37 (2017)
No. 4

数 学 杂 志
J. of Math. (PRC)

集装箱倒箱问题的模型与启发式算法研究

郭瑞智 [1], 史玛君 [1], 林昊堃 [2]
(1. 武汉大学数学与统计学院, 湖北 武汉 430072)
(2. 中国地质大学 (武汉) 环境学院, 湖北 武汉 430074)

摘 要: 本文研究了集装箱堆场中集装箱搬运的优化问题. 利用以 7 个倒箱落位步骤为核心的启发式算法, 建立轨道式龙门机取箱作业的数学模型, 获得了最小化倒箱量的方法. 推广了大规模倒箱问题的结果, 具有较好的实际意义.

关键词: 轨道式龙门机; 大规模倒箱; 倒箱量; 启发式算法
MR(2010) **主题分类号**: 03H05 **中图分类号**: O29
文献标识码: A **文章编号**: 0255-7797(2017)04-0805-06

1 引言

随着经济全球化的不断深入, 我国的进出口贸易不断增长, 随之带来港口集装箱吞吐量的急剧增长, 尤其在上海、广州、重庆等这样的大港, 业务量更是几年前的好几倍. 然而由于整体规划以及资金的考量, 集装箱堆场码头的面积和工作区域并没有明显的扩大. 如何在有限的条件下提高码头的运作效率, 便成为了当务之急.

在过去十几年, 集装箱吞吐量并不大的时候, 码头多采用轮胎式龙门机作为调运集装箱的起重机. 该起重机起重量、跨距均较小, 移动速度快, 方便灵活, 很适合中小码头的运作. 随着科技的发展以及集装箱吞吐量的增长, 轮胎式龙门机的弊端也显现出来: 无法满足大贝位垛堆集装箱的移动, 运行成本、维修成本较高, 轨道式龙门机应运而生. 比起轮胎式龙门机跨中只能堆放 6 列集装箱, 轨道式龙门机场地利用率更高, 跨中一般可堆放 8–15 列集装箱. 因此越来越多的码头开始采用该起重机作为调运集装箱的工具.

为了提高码头的运作效率, 除了采用更为先进的设备外, 减少提箱过程中集装箱倒箱次数也是一个重要的因素. 在集装箱堆场中, 竖直摆放的一列集装箱叫作一个栈, 并排摆放的几列栈称为一个贝. 若将要提出发箱的集装箱不在栈的最高层, 必须先把积压在其上的所有集装箱移动到其他栈, 此移动的过程称为倒箱. 许多实例表明, 贝位内集装箱数量越多, 倒箱就越多. 倒箱给提取集装箱装船带来了很多不必要的工作, 浪费了大量的时间和金钱精力. 为了解决这一困扰码头工作人员许久的难题, 不少学者开始研究它.

1997 年 Kim [1] 提出倒箱量估计方法, 研究了码头进口箱区堆存高度与倒箱量之间的关系, 并针对不同进口箱到达模式, 建立了最小化期望倒箱量为目标的数学模型; 2006 年 Kim 和 Hong [2] 利用分支定界和启发式算法研究了提箱过程中翻到集装箱落箱位置的确定问题; 范磊 [3] 建立了轮胎式龙门机取箱作业数学模型, 基于 6 条倒箱落位原则, 以倒箱量最少为目标, 运用启发式算法对模型进行求解; 徐亚 [4] 对阻塞箱落箱位置的确定问题以及如何减少二

*收稿日期: 2016-12-21 接收日期: 2017-02-27
作者简介: 郭瑞智 (1992–), 男, 湖南衡阳, 硕士, 主要研究方向: 最优化理论、算法及其应用.

超星·期刊

次倒箱进行了研究, 提出了一种启发式算法 H 及其改进算法 IH; 金鹏 [5] 以最小化翻箱总次数为优化目标, 运用多人爬山算法进行求解.

本文研究了贝位内的取箱作业问题, 给每一个箱子赋予一个优先级别, 数字越小的集装箱越优先被提取出贝位. 在设计启发式算法步骤时, 创新地引入阻塞箱与栈内优先级最高的箱子的差的绝对值这一概念, 并且将空栈单独列出作为考量. 通过大规模算例验证, 算法具有较好的可行解和时效性.

2 问题描述

贝位的初始状态和各集装箱的发箱顺序已知, 数字越小说明发箱优先级越高, 如顺序为 1 的集装箱是最先发箱的. 问题转化为按照各箱子发箱的优先级别依次取箱, 如何使得整个发箱过程倒箱量最少.

根据文献 [8] 中所提到的重庆寸滩港每一个贝位有 20 栈 (2 栈作为倒箱缓冲, 实际装箱为 18 栈), 5 层, 因此本文算例为 20 栈, 6 层, 稍大于实际规模. 贝位初始状态见表 1, 其中数字表示集装箱发箱顺序.

表 1: 贝位初始状态

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

3 取箱作业数学模型

3.1 模型假设

(1) 贝位初始状态和集装箱发箱顺序已知;
(2) 倒箱过程只在一个贝位内进行, 且仅对待提箱上方的阻塞箱进行倒箱操作;
(3) 不允许有新的集装箱进入贝位;
(4) 层数为 m 时, 贝位内至少有 $m-1$ 个空位留作倒箱使用.

3.2 符号说明

(i,j): 贝内第 i 层第 j 栈箱位, $i=1,2,\cdots,m; j=1,2,\cdots,n$, 贝位由 m 层 n 栈的集装箱组成, 最上层为第 1 层, 最左栈为第 1 栈, 以此类推; $B_{m\times n}$: 箱子优先级矩阵. 矩阵中各元素代表该位置上箱子被取出的优先级, 特别地, 0 元素表示空位; $b(k,i,j)$: 第 k 次搬移箱子前, 位于贝内第 i 层和第 j 栈的箱子的优先级, $k=1,2,\cdots,K; i=1,2,\cdots,m; j=1,2,\cdots,n$. $b(k,i,j)=0$ 表示空位;

$$p(k,i,j)=\begin{cases}1, & \text{若第 k 次搬移时某个箱子被放入到第 i 层第 j 栈,}\\ 0, & \text{否则;}\end{cases}$$

$$t(k,i,j)=\begin{cases}1, & \text{若第 k 次搬移时第 i 层第 j 栈的箱子被提出发箱,}\\ 0, & \text{否则.}\end{cases}$$

超星·期刊

3.3 模型的建立

3.3.1 模型的建立

$$F = \min \sum_{k=1}^{K} \sum_{i=1}^{m} \sum_{j=1}^{n} p(k,i,j). \tag{3.1}$$

式 (3.1) 表示倒箱数量最少.

3.3.2 约束条件

$$\begin{cases} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} p(k,i,j) + \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} t(k,i,j) = 1, k = 1,2,\cdots,K, & (3.2) \\ \text{如果 } b(k,i-1,j) = 0 \ \& \ b(k,i,j) \neq 0, \text{则 } p(k,i-1,j) \leqslant 1, \\ i = 2,\cdots,m; j = 1,\cdots,n; k = 1,\cdots,K, & (3.3) \\ \text{如果 } b(k,i-1,j) = 0 \ \& \ b(k,i,j) \neq 0, \text{则 } t(k,i,j) \leqslant 1, \\ i = 2,\cdots,m; j = 1,\cdots,n; k = 1,\cdots,K, & (3.4) \\ \sum\limits_{k=1}^{K} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} p(k,i,j) + \sum\limits_{k=1}^{K} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} = K, k = 1,2,\cdots,K, & (3.5) \\ b(K+1,i,j) = 0, i = 1,2,\cdots,m; j = 1,2,\cdots,n, & (3.6) \\ p(k,i,j) = 0 \text{ 或 } 1; t(k,i,j) = 0 \text{ 或 } 1, \\ i = 2,\cdots,m; j = 1,\cdots,n; k = 1,\cdots,K. & (3.7) \end{cases}$$

式 (3.2) 表示每次搬移, 仅有一个箱子被移动, 要么一个箱子被提出, 要么一个箱子被放入; 式 (3.3) 表示第 k 次搬移前箱位 (i,j) 上有箱子而 $(i-1,j)$ 上无箱子, 第 k 次才可能将箱子搬移到 $(i-1,j)$; 式 (3.4) 表示第 k 次搬移前箱位 (i,j) 上有箱子而 $(i-1,j)$ 上无箱子, 第 k 次才可能将 (i,j) 箱位的箱子提出发箱; 式 (3.5) 表示箱子搬移总次数为 K; 式 (3.6) 表示所有箱子均被发箱; 式 (3.7) 表示变量属性.

4 启发式算法

步骤 1 挑选其余栈中未到额定层数的空箱位作为翻出阻塞箱的堆存箱位.

步骤 2 步骤 1 成立时, 若空箱位所在栈内现有箱子优先级比预移入阻塞箱的都低, 则该空箱位可作为翻出箱的堆存箱位. 若这样的空箱位不唯一, 则计算阻塞箱与栈内优先级最高的箱子的差的绝对值, 选取绝对值最小的那一栈的空箱位作为翻出箱的堆存箱位.

步骤 3 步骤 1 成立但步骤 2 不成立时, 若空箱位所在栈内优先级最高的箱子上方无阻塞箱, 则该空箱位为翻出箱的堆存箱位. 若这样的空箱位不唯一, 则计算阻塞箱与栈内优先级最高的箱子的差的绝对值, 选取绝对值最小的那一栈的空箱位作为翻出箱的堆存箱位.

步骤 4 步骤 1 成立但步骤 2 和 3 不成立时, 若空箱位所在栈内优先级最高的箱子上方只有 1 个阻塞箱, 则该空箱位为翻出箱的堆存箱位. 若这样的空箱位不唯一, 则计算阻塞箱与

超星·期刊

栈内优先级最高的箱子的差的绝对值, 选取绝对值最小的那一栈的空箱位作为翻出箱的堆存箱位.

步骤 5 只有步骤 1 成立时, 若空箱位所在栈内优先级最高的箱子上方有 2 个阻塞箱, 则该空箱位为翻出箱的堆存箱位. 若这样的空箱位不唯一, 则计算阻塞箱与栈内优先级最高的箱子的差的绝对值, 选取绝对值最小的那一栈的空箱位作为翻出箱的堆存箱位.

步骤 6 只有步骤 1 成立时, 若空箱位所在栈内优先级最高的箱子上方有 3 个阻塞箱, 则该空箱位为翻出箱的堆存箱位. 若这样的空箱位不唯一, 则计算阻塞箱与栈内优先级最高的箱子的差的绝对值, 选取绝对值最小的那一栈的空箱位作为翻出箱的堆存箱位.

步骤 7 只有步骤 1 成立时, 若空箱位所在栈内优先级最高的箱子上方有 4 个阻塞箱, 则该空箱位为翻出箱的堆存箱位. 若这样的空箱位不唯一, 则计算阻塞箱与栈内优先级最高的箱子的差的绝对值, 选取绝对值最小的那一栈的空箱位作为翻出箱的堆存箱位.

对于空栈的处理: 在执行上述任何一步时, 若出现空栈, 则选定目前非空栈中最上层箱子优先级最低的那一个. 若该箱子不是该栈优先级最高的, 则将其移入空栈中, 再继续执行上述 7 个步骤; 否则, 把将要移动的阻塞箱放入空栈中, 再继续执行上述 7 个步骤.

5 算例与结果分析

运用上述启发式算法, 根据表 1 的初始贝位状态, 对该 20 栈、6 层的较大规模的集装箱堆场进行倒箱操作, 可以得到倒箱优化方案, 见表 2.

表 2: 20*6 集装箱堆场倒箱优化方案

倒箱次数	1	2	3	4	5	6	7	8	9	10	11	12	13
阻塞箱顺序	86	27	22	67	100	8	99	79	40	50	21	88	78
倒箱前位置	2 层 7 栈	3 层 7 栈	4 层 7 栈	5 层 7 栈	1 层 20 栈	2 层 8 栈	3 层 8 栈	4 层 8 栈	2 层 11 栈	2 层 14 栈	3 层 14 栈	4 层 14 栈	2 层 4 栈
倒箱后位置	1 层 19 栈	2 层 12 栈	1 层 12 栈	1 层 9 栈	6 层 7 栈	1 层 13 栈	5 层 7 栈	4 层 7 栈	5 层 8 栈	3 层 7 栈	3 层 11 栈	2 层 7 栈	5 层 14 栈

倒箱次数	14	15	16	17	18	19	20	21	22	23	24	25	26
阻塞箱顺序	23	42	56	43	24	73	99	16	36	47	87	33	97
倒箱前位置	2 层 3 栈	3 层 3 栈	4 层 3 栈	3 层 1 栈	4 层 1 栈	2 层 13 栈	1 层 16 栈	2 层 16 栈	3 层 16 栈	4 层 16 栈	2 层 2 栈	2 层 6 栈	3 层 6 栈
倒箱后位置	3 层 4 栈	1 层 7 栈	4 层 8 栈	4 层 14 栈	2 层 15 栈	5 层 3 栈	3 层 14 栈	1 层 20 栈	3 层 8 栈	3 层 13 栈	5 层 16 栈	2 层 8 栈	2 层 13 栈

倒箱次数	27	28	29	30	31	32	33	34	35	36	37	38	39
阻塞箱顺序	63	96	71	76	81	99	67	86	91	39	60	92	74
倒箱前位置	4 层 6 栈	2 层 17 栈	3 层 17 栈	4 层 17 栈	5 层 17 栈	3 层 14 栈	1 层 9 栈	1 层 19 栈	2 层 20 栈	3 层 20 栈	4 层 20 栈	2 层 10 栈	3 层 10 栈
倒箱后位置	4 层 3 栈	5 层 6 栈	4 层 6 栈	3 层 6 栈	3 层 3 栈	6 层 17 栈	2 层 6 栈	5 层 17 栈	4 层 17 栈	3 层 14 栈	2 层 3 栈	1 层 6 栈	3 层 17 栈

超星·期刊

倒箱次数	40	41	42	43	44	45	46	47	48	49	50	51	52
阻塞箱顺序	94	90	38	82	65	57	97	64	31	92	85	45	52
倒箱前位置	1 层 18 栈	2 层 18 栈	3 层 18 栈	4 层 18 栈	3 层 2 栈	5 层 2 栈	2 层 13 栈	4 层 4 栈	5 层 4 栈	1 层 6 栈	3 层 12 栈	3 层 9 栈	4 层 9 栈
倒箱后位置	2 层 17 栈	1 层 3 栈	2 层 14 栈	5 层 20 栈	1 层 17 栈	4 层 16 栈	6 层 2 栈	5 层 2 栈	5 层 1 栈	6 层 4 栈	5 层 4 栈	4 层 12 栈	4 层 10 栈

倒箱次数	53	54	55	56	57	58	59	60	61	62	63	64	65
阻塞箱顺序	83	59	72	90	89	82	41	80	75	56	88	51	77
倒箱前位置	3 层 19 栈	4 层 19 栈	5 层 19 栈	1 层 3 栈	3 层 15 栈	5 层 20 栈	4 层 11 栈	4 层 15 栈	5 层 15 栈	4 层 8 栈	2 层 7 栈	3 层 5 栈	4 层 5 栈
倒箱后位置	4 层 4 栈	4 层 2 栈	3 层 4 栈	6 层 1 栈	6 层 9 栈	6 层 19 栈	2 层 5 栈	5 层 19 栈	4 层 19 栈	3 层 16 栈	6 层 15 栈	3 层 10 栈	5 层 15 栈

倒箱次数	66	67	68	69	70	71	72	73	74	75	76	77	
阻塞箱顺序	70	78	61	84	62	87	81	94	73	76	91	96	
倒箱前位置	5 层 5 栈	5 层 14 栈	5 层 12 栈	4 层 13 栈	5 层 10 栈	5 层 16 栈	3 层 3 栈	2 层 17 栈	5 层 3 栈	3 层 6 栈	4 层 17 栈	5 层 6 栈	
倒箱后位置	2 层 4 栈	6 层 5 栈	1 层 6 栈	6 层 12 栈	5 层 11 栈	6 层 10 栈	6 层 14 栈	6 层 16 栈	2 层 17 栈	4 层 15 栈	5 层 4 栈	5 层 2 栈	

总倒箱次数为 77, 算法耗时 0.037s. 为了进一步验证上述启发式算法的高效性, 对文献 [3] 中的算例进行计算, 贝位初始状态见表 3, 计算结果见表 4.

表 3: 6 层 6 栈的贝位初始状态

					20
	1				22
	11		9	10	21
12	13	14	7	8	23
3	5	6	16	17	24
15	2	4	18	19	25

表 4: 6*6 集装箱堆场倒箱优化方案

倒箱次数	1	2	3	4	5	6	7	8	9	10	11
阻塞箱顺序	11	13	5	12	14	6	13	9	11	10	22
倒箱前位置	3 层 2 栈	4 层 2 栈	5 层 2 栈	4 层 1 栈	4 层 3 栈	5 层 3 栈	2 层 4 栈	3 层 4 栈	2 层 5 栈	3 层 5 栈	2 层 6 栈
倒箱后位置	2 层 5 栈	2 层 4 栈	1 层 4 栈	6 层 2 栈	5 层 1 栈	1 层 5 栈	4 层 1 栈	5 层 2 栈	3 层 1 栈	2 层 1 栈	6 层 1 栈

总倒箱次数为 11, 与文献 [3] 结果一样, 但是系统记录运行时间为 0.018s, 比文献 [3] 的 0.094s 快了几倍, 因此本文的启发式算法耗时更短, 时效性更好.

6 结论

超星·期刊

本文以轨道式龙门机在贝位内取箱作业过程中倒箱量最少为目标, 建立了取箱作业的数学模型, 运用 MATLAB 编译的启发式算法对模型进行求解. 相比于同类文章只研究了小规模的集装箱倒箱问题, 本文研究的是 20 栈、6 层的大规模问题, 更加贴近实际, 具有较好的现实意义; 单独将空栈拿出来考虑, 有利于减少二次、三次倒箱数量, 对于减少总倒箱量有很大帮助; 针对启发式算法步骤中出现候选栈不唯一的情况, 本文多增加了一个约束条件, 从而确保候选栈的唯一性, 大大减少了运算量和随机性. 本文通过两个规模不同的算例对算法进行了验证, 均给出了具体的倒箱较优方案. 算例 1 的结果对于生产实际中大港口的取箱作业有较好的帮助, 算例 2 的结果表明本文算法在时间上具有很大优势. 事实上, 由于假设 2 的存在, 集装箱倒箱实际上是一个 NP-hard 问题, 本文提出的启发式算法属于贪心算法范畴, 求得的解可能不是全局最优解, 下一步的研究重点将放到非限制变量上.

参 考 文 献

[1] Kim K H. Evaluation of the number of rehandles in container yards[J]. Comput. Indus. Engin. (S0360-8352), 1997, 32(4): 701–711.

[2] Kim K H, Gyu-Pyo Hong. A heuristic rule for relocating blocks[J]. Comp. Oper. Res., 2006, 33(4): 940–954.

[3] 范磊, 梁承姬. 堆场取箱作业中倒箱问题的启发式算法研究 [J]. 重庆交通大学学报 (自然科学版), 2014, 33(1): 133–138.

[4] 徐亚, 陈秋双, 龙磊等. 集装箱倒箱问题的启发式算法研 [J]. 系统仿真学报, 2008, 20(14): 3666–3670.

[5] 金鹏, 黄有方, 严伟. 位内集装箱翻箱操作的启发式优化 [J]. 上海海事大学学报, 2009, 30(4): 13–17.

[6] 张维英, 林焰, 纪卓尚, 吴毅刚. 出口集装箱堆场取箱作业优化模型研究 [J]. 武汉理工大学学报 (交通科学与工程版), 2006, 30(2): 314–317.

[7] 张德文, 谢琛, 陈丽昕. 轨道式集装箱门式起重机的技术分析 [J]. 水运科学研究, 2006, 6(2):35–41.

RESEARCH ON THE MODEL AND HEURISTIC ALGORITHM OF CONTAINER RELOCATION PROBLEM

GUO Rui-zhi[1], SHI Ma-jun[1], LIN Hao-kun[2]

(*1.School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China*)

(*2.School of Environmental Studies, China University of Geosciences, Wuhan 430074, China*)

Abstract: In this paper, we study the optimization of container handling in container yard. Based on the heuristic algorithm with seven choosing locations steps, a mathematical model for railroad gantry crane's retrieving operation is established, and the method of minimizing the relocation numbers is obtained. This paper promotes the result of large-scale relocation problem which has good practical significance.

Keywords: railroad gantry crane; large-scale relocation; relocation number; heuristic algorithm

2010 MR Subject Classification: 03H05