

集装箱码头堆场翻箱与外集卡提箱顺序同步优化方法

冯媛君, 曾庆成, 李厚源, 杨忠振

(大连海事大学 交通运输管理学院, 辽宁 大连 116026)

摘要: 针对集装箱码头提箱作业过程中, 由于外集卡的提箱顺序与目标箱在堆场的堆存位置不匹配导致大量翻箱这一难题, 以码头的作业成本和外集卡的延误成本之和最小为目标, 建立堆场翻箱与外集卡提箱顺序同步优化模型, 优化外集卡的提箱顺序、龙门吊的任务分配以及翻箱方案。设计基于动态规划的启发式算法求解模型, 并利用算例对模型与算法的有效性进行了验证。结果表明: 与目前码头普遍采用的提箱方式相比, 通过调整外集卡提箱顺序并同时优化翻箱方案以及龙门吊的任务分配可以降低堆场翻箱率, 减少龙门吊的移动成本, 从而节省提箱作业的总成本。

关键词: 物流工程; 码头翻箱方法; 优化算法; 集装箱码头

中图分类号: U691 **文章标识码:** A **文章编号:** 1007-3221(2017)05-0110-09 **doi:** 10.12005/orms.2017.0116

Method for Integrated Optimizing of Rehandling and Retrieving Sequence for Inbound Containers in Container Terminals

FENG Yuan-jun, ZENG Qing-cheng, LI Hou-yuan, YANG Zhong-zhen

(School of Transportation Management, Dalian Maritime University, Dalian 116026, China)

Abstract: During the operation of retrieving containers in container terminals, the retrieving sequence of inbound containers is not matched with the location of the target containers, which results in huge amounts of rehandling. In order to address this problem, an integrated optimizing model of rehandling and retrieving sequence for inbound containers is developed to optimize the retrieving sequence, yard crane assignment and rehandling scheme. The objective is to minimize the operation cost of container terminals and the lateness cost of external trucks. To solve the model, a dynamic programming-based heuristic is designed. Numerical experiments are provided to illustrate the validity of the proposed model and algorithms. The results indicate that the integrated optimizing of retrieving sequence, rehandling scheme and yard crane assignment can decrease the rehandling ratio and the moving cost of yard cranes, and thus can save the total cost of retrieving operation.

Key words: logistics engineering; container rehandling method of terminals; optimization algorithms; container terminals

0 引言

进口箱的提箱作业是集装箱码头堆场的主要作业环节之一。由于外集卡的提箱顺序与目标箱在堆场的堆存位置顺序不匹配, 提箱过程中产生大量翻箱, 这不仅增加外集卡在码头的周转时间, 降低堆场提箱作业效率^[1], 而且增加码头作业成本。因此, 如何减少翻箱成为国内外学者及码头管理者

共同关注的问题。目前, 国内外学者针对翻箱策略进行了大量的研究, 但是翻箱受外集卡到达与堆场作业调度的共同影响, 单独考虑翻箱策略不能从根本上解决翻箱问题。近年来, 集卡预约在国内外港口相继实施^[2~5], 码头通过集卡预约信息可以提前获得外集卡的到港时间, 这为调整外集卡的提箱顺序、降低翻箱率^[6]提供了条件。同时优化翻箱策略与外集卡的提箱顺序为减少堆场翻箱提供了新思路。

收稿日期: 2015-10-21

基金项目: 国家自然科学基金项目(71370137); 大连市杰出青年人才项目(2015J12JH202); 中央高校基本科研业务费专项资金项目(3132016302, 3132016306)

作者简介: 曾庆成(1978-), 男, 博士, 教授, 博士生导师, 研究方向: 港口与物流系统优化; 冯媛君(1992-), 女, 硕士研究生, 研究方向: 港口与物流系统优化。

提箱作业环节涉及多个操作过程,其效率不仅受翻箱量影响,还受龙门吊的任务分配以及移动路径的影响,而外集卡的提箱顺序影响龙门吊的移动路径与作业成本。同时,龙门吊作业顺序与翻箱落位相互影响,而且堆场堆存状态、目标箱所需的作业时间随翻箱实时改变,这给模型的构建和求解带来了困难。

为解决上述问题,本文在已有翻箱问题和龙门吊调度问题研究的基础上,考虑翻箱成本、龙门吊移动成本与外集卡等待成本,建立堆场翻箱与外集卡提箱顺序同步优化模型,优化翻箱方案、龙门吊的任务分配与作业顺序。

1 国内外研究现状及分析

堆场作业是制约码头效率的关键因素之一,国内外学者开展了大量的研究,开发了一系列模型与算法。一方面,一些学者致力于研究如何优化翻倒箱落位以降低翻箱率。由于翻箱问题是 NP-hard 问题^[7],大多数研究设计启发式算法来求解翻箱落位,典型研究如, Kim 和 Hong^[8]、徐亚等^[9]、Lee 和 Chao^[10]、Lee 和 Hsu^[11]、Wan 等^[12] 和 Tang 等^[13]。其中, Lee 和 Hsu^[11]、Wan 等^[12]、Tang 等^[13] 以翻箱量最小为目标建立了整数规划模型。另外,一些研究应用动态规划方法求解翻箱方案,如靳志宏等^[1]、Caserta 等^[14]。由于翻箱过程涉及龙门吊大车与小车的移动,而龙门吊的移动时间也会影响堆场的作业效率,因此,一些研究同时关注翻箱量与龙门吊的移动时间。如 Unluyurt 和 Aydin^[15] 以最小化翻箱时间与龙门吊水平移动时间为目标获得最优翻箱方案。Lee^[16] 以最小化翻箱数量与龙门吊工作时间为目标,开发了一种三阶段启发式算法来优化单台龙门吊的取箱作业顺序。边展等^[17] 针对多贝位、大规模的出口箱的取箱问题,以最少的翻箱次数与最短的龙门吊工作时间为目标,优化两台龙门吊的取箱作业集合。

另一方面,一些学者致力于研究龙门吊调度问题,主要模型可分为两类。一类是单台龙门吊调度模型(Kim 等^[18]; Ng 和 Mak^[19]), 另一类是多台龙

门吊调度模型(Ng^[20]; Li 等^[21]; Li 等^[22]; He 等^[23]; Chang 等^[24]; He 等^[25])。由于多台龙门吊作业时彼此之间不能干扰,多台龙门吊调度模型考虑的约束更复杂。如, Li 等^[21] 考虑了龙门吊之间的固定作业距离以及同时提箱与交箱操作。Li 等^[22] 在文献[21]的基础上,考虑更多的约束条件(龙门吊在前后两个作业任务之间的移动时间等约束),将离散时间模型转化为连续时间模型。此外,考虑龙门吊的作业效率与能源消耗之间的平衡, He 等^[25] 将龙门吊调度问题转化为带软时间窗的车辆路径问题,以作业任务延误和龙门吊能源消耗最小为目标建立混合整数规划模型。

上述研究多数是分别优化翻箱与龙门吊调度,文献[15]和文献[17]考虑了龙门吊作业时间,但没有优化龙门吊任务分配与作业顺序。本文在已有研究的基础上,建立堆场翻箱与外集卡提箱顺序同步优化模型,在已知集卡到港信息的情况下,通过优化外集卡的提箱顺序、龙门吊的任务分配以及翻箱落位来降低提箱总成本。

2 堆场翻箱与外集卡提箱顺序同步优化模型

2.1 问题描述

龙门吊服务外集卡提箱的过程为: 龙门吊在贝位间移动→翻箱→提取目标箱。如图1,如果按照外集卡到达顺序进行提箱作业,目标箱5是目标箱1的阻碍箱,为避免翻箱,可以对提箱顺序进行调整,即优化龙门吊任务的作业顺序。对于无法避免的翻箱,需要为其选择最佳的落箱位以避免二次翻箱。同时,为减少龙门吊移动,需要优化龙门吊的任务分配以及任务的作业顺序。然而,调整提箱顺序可能增加某些外集卡提箱时间。根据上述问题的特点,本文以提箱总成本(翻箱成本、龙门吊移动成本以及外集卡的延误成本)最小为目标,考虑翻箱时间、龙门吊移动时间、外集卡最长在港时间等约束,构建堆场翻箱与外集卡提箱顺序同步优化模型,同时优化外集卡的提箱顺序,翻倒箱的落箱位以及龙门吊的任务分配。

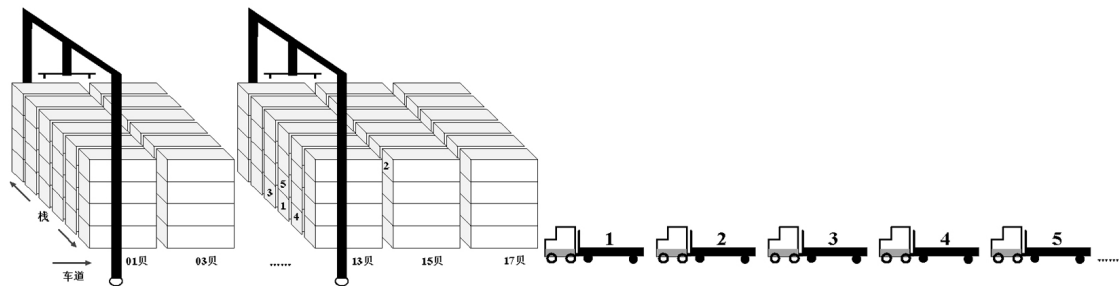


图1 外集卡提箱示意图

2.2 模型构建

为构建堆场翻箱与外集卡提箱顺序同步优化模型,假设:(1)计划期内外集卡到港时间已知,且到港时间视为进入堆场内的时间;(2)翻箱只在同一贝位中进行,且只在提箱时发生;(3)翻箱时间只与翻箱数量有关,单次翻箱耗时为单次提箱耗时的2倍。

模型参数定义为: α 表示一个街区中贝位的数量,每个贝按顺序从1到 α 编号; P 表示每一贝中层的数量,每一层由低到高从1到 P 编号; C 表示每一贝中栈的数量,每一栈按顺序从1到 C 编号; m 表示一个街区中龙门吊的数量,每个龙门吊按顺序从1到 m 编号(与贝的编号方向一致); n 表示待提目标箱的数量; a_i 表示提取目标箱 i 的外集卡到港时间, n 个目标箱按照 $a_i \leq a_{i+1}$ 的顺序编号; d_i 表示提取目标箱 i 的外集卡的预计离港时间; e_i 表示提取目标箱 i 的外集卡的预计最晚离港时间; l_i 表示目标箱 i 所在的贝位; X_{icp} 表示初始状态时集装箱所在的层和列。如果集装箱 i 堆存在第 c 列的第 p 层, $X_{icp} = 1$;否则, $X_{icp} = 0$ 。 h_1 表示单次翻箱耗时; h_2 表示龙门吊移动一个贝位耗时; h_3 表示龙门吊在贝位间移动时加速和减速耗时之和; c_1 表示单次翻箱成本; c_2 表示龙门吊移动一个贝位的成本; c_3 表示外集卡单位延误成本。将所有目标箱提箱作业完成时间的上界划分为 θ 个时间段,每个时间段的长度为 h_1, h_2, h_3 与外集卡到达时间计量单位中的最小值。

决策变量定义为:如果目标箱 i 在 j 之前(紧前)被龙门吊 k 服务, $S_{ijk} = 1$;否则, $S_{ijk} = 0$ 。如果龙门吊 k 在 t 时段在 b 贝位作业, $Y_{kbt} = 1$;否则, $Y_{kbt} = 0$ 。 $Y_{kb0} = 1$ 表示龙门吊 k 的初始位置为 b_0 ;对于 $b \neq b_0$,令 $Y_{kb0} = 0$ 。如果龙门吊 k 在 t 时段完成目标箱 i 的提箱作业, $Z_{ikt} = 1$;否则, $Z_{ikt} = 0$ 。 t_{1i} 表示提目标箱 i 时翻箱耗时; t_{2i} 表示提目标箱 i 时龙门吊在贝位间移动耗时; r_i 表示龙门吊开始服务目标箱 i 的时间段。如果 t 时段集装箱 i 在第 c 列的第 p 层, $x_{icp} = 1$;否则 $x_{icp} = 0$ 。如果集装箱 i 是 t 时段目标箱的阻碍箱, $y_{ii} = 1$;否则 $y_{ii} = 0$;如果集装箱 i 和 j 都是 t 时段目标箱的阻碍箱,且在翻倒前 j 的位置比 i 高, $w_{ij} = 1$;否则 $w_{ij} = 0$ 。

于是,堆场翻箱与外集卡提箱顺序同步优化模型可以表示为:

$$\min c_3 \cdot \sum_{i=1}^n \sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} (t \cdot Z_{ikt} - d_i) + c_1 \cdot \sum_{i=1}^n \sum_{j=1}^n y_{r_{ij}} + c_2 \cdot \sum_{i=2}^n \sum_{j=1}^n \sum_{k=1}^m S_{jik} \cdot |l_j - l_i| \quad (1)$$

$$\text{s. t. } t_{1i} = h_1 \cdot \sum_{j=1}^n y_{r_{ij}}, \quad 1 \leq i \leq n \quad (2)$$

$$t_{2i} = h_2 \cdot \sum_{i=2}^n \sum_{j=1}^n \sum_{k=1}^m S_{jik} \cdot |l_j - l_i| + h_3, 1 \leq i \leq n \quad (3)$$

$$r_i + t_{2i} \geq a_i, 1 \leq i \leq n \quad (4)$$

$$\sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt} = r_i + t_{2i} + t_{1i} + \frac{h_1}{2}, 1 \leq i \leq n \quad (5)$$

$$\sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt} \leq e_i, 1 \leq i \leq n \quad (6)$$

$$\sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} Z_{ikt} = 1, 1 \leq i \leq n \quad (7)$$

$$\sum_{b=1}^{\alpha} Y_{kbt} = 1, 1 \leq k \leq m, 1 \leq t \leq \theta \quad (8)$$

$$\sum_{j=0}^{t_1 + \frac{h_1}{2}} Y_{kl(t-j)} - (t_{1i} + \frac{h_1}{2}) - 1 \geq M(Z_{ikt} - 1), 1 \leq k \leq m, 1 \leq i \leq n, \frac{h_1}{2} \leq t \leq \theta \quad (9)$$

$$M(1 - Y_{kbt}) \geq \sum_{j=b}^{\alpha} Y_{(k-1)jt}, 2 \leq k \leq m, 1 \leq b \leq \alpha, 1 \leq t \leq \theta \quad (10)$$

$$\sum_{t=\frac{h_1}{2}}^{\theta} t(Z_{jkt} - Z_{ikt}) \geq \frac{h_1}{2} + t_{1i} + t_{2i} - M(1 - S_{ijk}), 1 \leq i \leq n, 1 \leq j \leq n, i \neq j, 1 \leq k \leq m \quad (11)$$

$$\sum_{t=\frac{h_1}{2}}^{\theta} (Z_{jkt} + Z_{ikt}) - 1 \leq S_{ijk} + S_{jik}, 1 \leq i \leq n, 1 \leq j \leq n, i \neq j, 1 \leq k \leq m \quad (12)$$

$$(1 - \sum_{p=1}^P x_{r_i, icp})P + y_{r_i, j} \geq (\sum_{p=1}^P p x_{r_i, jcp} - \sum_{p=1}^P p x_{r_i, icp})/P, 1 \leq i < j \leq n, 1 \leq c \leq C \quad (13)$$

$$(\sum_{p=1}^P p x_{r_i, icp} - \sum_{p=1}^P p x_{r_i, jcp})/P \leq 1 - y_{r_i, j}, 1 \leq i < j \leq n, 1 \leq c \leq C \quad (14)$$

$$\sum_{c=1}^C \sum_{p=1}^P x_{ticp} = 1, 1 \leq i \leq n, 1 \leq t \leq \theta \quad (15)$$

$$\sum_{i=1}^n x_{ticp} \leq 1, 1 \leq t \leq \theta, 1 \leq c \leq C, 1 \leq p \leq P \quad (16)$$

$$\sum_{i=1}^n x_{ticp} \leq \sum_{i=1}^n x_{tic, p-1}, 1 \leq t \leq \theta, 1 \leq c \leq C, 2 \leq p \leq P \quad (17)$$

$$\sum_{p=1}^P x_{r_i,j} \sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt,jcp} \leq 2 - y_{r_i,j} - \sum_{p=1}^P x_{r_i,icp},$$

$$1 \leq i < j \leq n-1, 1 \leq c \leq C \quad (18)$$

$$2 - y_{r_s,i} - y_{r_s,j} + w_{r_s,ij} \geq \left(\sum_{c=1}^C \sum_{p=1}^P p x_{r_s,jcp} - \sum_{c=1}^C \sum_{p=1}^P p x_{r_s,icp} \right) / P,$$

$$1 \leq s < i \leq n, 1 \leq s < j \leq n, i \neq j \quad (19)$$

$$y_{r_s,i} + y_{r_s,j} + w_{r_s,ij} \leq 3 + \left(\sum_{c=1}^C \sum_{p=1}^P p x_{r_s,jcp} - \sum_{c=1}^C \sum_{p=1}^P p x_{r_s,icp} \right) / P,$$

$$1 \leq s < i \leq n, 1 \leq s < j \leq n, i \neq j \quad (20)$$

$$w_{r_s,ij} \leq y_{r_s,i}, \quad 1 \leq s < i \leq n, 1 \leq s < j \leq n, i \neq j \quad (21)$$

$$w_{r_s,ij} \leq y_{r_s,j}, \quad 1 \leq s < i \leq n, 1 \leq s < j \leq n, i \neq j \quad (22)$$

$$\sum_{p=1}^P p x_{r_i,jcp} \sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt,icp} - \sum_{p=1}^P p x_{r_i,jcp} \sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt,jcp} \geq -P(1 - w_{r_s,ij})$$

$$-P(1 - y_{r_s,i}) - P(1 - y_{r_s,j}) - P(1 - \sum_{p=1}^P x_{r_i,jcp} \sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt,icp}),$$

$$1 \leq s < i \leq n, 1 \leq s < j \leq n, i \neq j, 1 \leq c \leq C \quad (23)$$

$$x_{r_i,jcp} - x_{r_i,jcp} \geq -y_{r_i,j},$$

$$\sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt,jcp}$$

$$1 \leq i \leq n, 2 \leq j \leq n, 1 \leq c \leq C, 1 \leq p \leq P \quad (24)$$

$$x_{r_i,jcp} - x_{r_i,jcp} \geq -y_{r_i,j},$$

$$\sum_{k=1}^m \sum_{t=\frac{h_1}{2}}^{\theta} t \cdot Z_{ikt,jcp}$$

$$1 \leq i \leq n, 2 \leq j \leq n, 1 \leq c \leq C, 1 \leq p \leq P \quad (25)$$

$$x_{1icp} = X_{1icp}, \quad 1 \leq i \leq n, 1 \leq c \leq C, 1 \leq p \leq P \quad (26)$$

$$S_{ijk}, Y_{kbt}, Z_{ikt} \in \{0, 1\}, \quad 1 \leq i \leq n, 1 \leq j \leq n, i \neq j,$$

$$1 \leq k \leq m, 1 \leq b \leq \alpha, 1 \leq t \leq \theta \quad (27)$$

$$y_{si} \in \{0, 1\}, \quad 1 \leq s \leq \theta, 1 \leq i \leq n \quad (28)$$

$$w_{sij} \in \{0, 1\}, \quad 1 \leq s \leq \theta, 2 \leq i \leq n, 2 \leq j \leq n, i \neq j \quad (29)$$

$$x_{sicp} \in \{0, 1\}, \quad 1 \leq s \leq \theta, 1 \leq i \leq n, 1 \leq c \leq C, 1 \leq p \leq P \quad (30)$$

式(1)为目标函数,表示最小化外集卡的延误成本,翻箱成本和龙门吊的移动成本。式(2)表示提目标箱*i*的翻箱时间。式(3)表示为提目标箱*i*时龙门吊的移动时间(包括走行时间和加速、减速时间)。式(4)表示外集卡到港之后,龙门吊才开始为其服务。式(5)表示目标箱提箱*i*作业的完成时间等于龙门吊开始为其服务的时间加上翻箱时间和提箱时间。式(6)表示目标箱*i*的提箱作业完成时间不能超过其对应外集卡的预计最晚离港时间。式(7)表示对于每一个目标箱,其提箱作业的完成时间只有一个。式(8)表示每一个龙门吊在每一个时间段只能在一个贝位。式(9)表示在龙门吊*k*为目标箱*i*进行提箱服务期间内,一直处在目标箱所在贝位。式(10)表示龙门吊之间不能

交叉作业。式(11)表示由同一个龙门吊相继服务的两个目标箱的提箱作业完成时间之间的关系。式(12)表示由同一个龙门吊作业的目标箱所对应变量之间的关系。式(13)和(14)用来决定翻箱变量*y_{r_{i,j}}*。考虑堆存在同一列的两个集装箱*i*和*j*,*j*在*i*上面,式(2)变为*y_{r_{i,j}}* ≥ (*p_j* - *p_i*) / *P*。此时该式的右侧为小于1的正数,使得*y_{r_{i,j}}*取1,即为提取目标箱*i*,集装箱*j*需要被翻倒。在其他情况下,式(13)对*y_{r_{i,j}}*的取值没有任何约束,而式(14)使得*y_{r_{i,j}}*取0,即当提目标箱*i*时集装箱*j*不需要被翻倒。式(15)保证每个集装箱占据一个箱位。式(16)表示每个箱位至多被一个集装箱占据。式(17)防止集装箱悬空堆放。式(18)保证阻碍箱不能移到当前自己所在的栈。式(19)至式(22)根据两个阻碍箱的相对高度确定变量*w_{sij}*。式(23)确定翻倒后集装箱*i*,*j*的相对高度,使得翻倒后集装箱*i*和集装箱*j*的相对高度互换。式(24)和(25)保证不翻倒的集装箱在下一阶段的位置不变。式(26)用来初始化集装箱的堆存位置。式(27)至(30)定义了0-1决策变量。

3 求解算法

已有研究表明,龙门吊调度问题属于 NP-complete问题^[20],随着问题规模的增大,难以在有效的时间内获得问题的最优解,而考虑翻箱的多台龙门吊调度问题更加复杂。本文设计基于动态规划的启发式算法,算法总体流程如图2所示,其基本思路是将多台龙门吊的调度问题分解成多个单台龙门吊的调度问题,假设每台龙门吊的作业范围是几个连续的贝位,每个贝位只能被一台龙门吊服务。由于单台龙门吊的调度问题仍属于 NP-complete问题,因此,采用贪婪算法求解考虑翻箱的单台龙门吊调度问题。通过贪婪算法获得每台龙门吊作业范围内的提箱总成本,将其返回到动态规划中,以优化每台龙门吊的作业范围(贝位范围)、作业范围内目标箱的作业顺序以及翻倒箱落位。

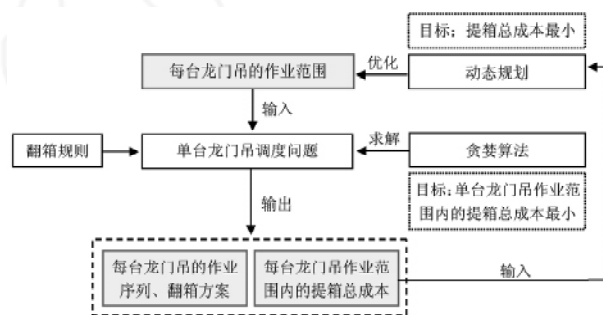


图2 基于动态规划的启发式算法流程图

(1) 龙门吊作业范围分配: 以 b_k 表示龙门吊 k 作业范围内的最小贝位号, $k=1, 2, \dots, m$; $g(i, j)$ 表示贝位 (i, j) 范围内所有目标箱的提箱总成本的最小和; $f(k, r)$ 表示龙门吊 $k, k+1, \dots, m$ 作业范围内所有目标箱的提箱总成本的最小和, 其中 $r=b_k$, $f(1, 1)$ 即表示所有目标箱提箱总成本的最小和。

龙门吊 $k, k+1, \dots, m$ 的作业范围分别是 $\{r, r+1, \dots, b_{k+1}-1\}, \{b_{k+1}, b_{k+1}+1, \dots, b_{k+2}-1\}, \dots, \{b_m, b_m+1, \dots, \alpha\}$ 。 b_2, b_3, \dots, b_m 将作业区域划分为了 m 个部分, 分别分配给 m 个龙门吊。通过式 (31)、(32) 求出使贝位 $(1, \alpha)$ 范围内所有目标箱的提箱总成本之和最小的 b_2, b_3, \dots, b_m , $f(1, 1)$ 即可确定。

$$f(m, r) = g(r, \alpha), \quad r = m, m+1, \dots, \alpha \quad (31)$$

$$f(k, r) = \min_{j=r+1, r+2, \dots, \alpha-m+k+1} \{f(k+1, j) + g(r, j-1)\},$$

$$k \leq r \leq \alpha - m + k, 1 \leq k \leq m \quad (32)$$

(2) 单台龙门吊调度优化: 采用贪婪算法求解 $g(i, j)$ 。对于作业范围是 $\{i, i+1, \dots, j\}$ 的考虑翻箱的单台龙门吊的调度问题, 其贪婪算法流程如下:

Step 1 令 $t=0$; 令 $l=i$, i 是作业范围为 $\{i, i+1, \dots, j\}$ 的龙门吊的初始位置。令 $\Psi = \{1, 2, \dots, n\}$, 表示该台龙门吊作业范围内未提走的集装箱的集合, n_i 表示该龙门吊作业范围内集装箱的数量。

Step 2 如果 $\Psi = \phi$, 停止; 否则, 针对 Ψ 内每个集装箱 q , 计算 t 时刻其阻碍箱的数量 n_q 。

Step 3 针对 Ψ 内每个集装箱 q , 计算其翻箱成本 C_1 、龙门吊由 l 移动到 l_q 的移动成本 C_2 。

Step 4 针对 Ψ 内每个集装箱 q , 计算其提箱作业的完成时间 t_q 。

Step 5 针对 Ψ 内每个集装箱 q , 计算 t_q 时刻, 已到港但未被服务的所有外集卡的延误成本 C_3 。

Step 6 针对 Ψ 内每个集装箱 q , 计算 C_1, C_2, C_3 之和 C , 将 C 最小的集装箱 \check{q} 作为下一个提取的目标箱。

Step 7 针对 Ψ 内每个集装箱 q , 若其预计最晚离开时间大于 t_q , 则令 $\check{q} = q$, 将集装箱 \check{q} 作为下一个提取的目标箱。

Step 8 如果 $n_{\check{q}} = 0$, 执行 Step13; 否则, 令 $n = 1$, 执行 Step9。

Step 9 如果 $n > n_{\check{q}}$, 执行 Step13。否则, 执行 Step10。

Step 10 对于 \check{q} 的第 n 个阻碍箱, 检查其是否存在第一类候选栈, 如果不存在, 执行 Step11; 如果存在一个第一类候选栈, 则将其作为第 n 个阻碍箱的落箱位。否则, 选出每个第一类候选栈中提箱顺

序最早的集装箱, 将这些集装箱的提箱顺序进行排序, 然后在其中选择提箱顺序最早的集装箱所在的栈作为第 n 个阻碍箱的落箱位。 $n = n + 1$, 执行 Step9。

Step 11 对于 \check{q} 的第 n 个阻碍箱, 检查其是否存在空栈, 如果不存在, 执行 Step12; 如果存在一个空栈, 则将其作为第 n 个阻碍箱的落箱位。否则, 选择最左侧的空栈作为第 n 个阻碍箱的落箱位。 $n = n + 1$, 执行 Step9。

Step 12 选出每个第二类候选栈中提箱顺序最早的集装箱, 将这些集装箱的提箱顺序进行排序, 然后在其中选择提箱顺序最早的集装箱所在的栈作为第 n 个阻碍箱的落箱位。 $n = n + 1$, 执行 Step9。

Step 13 将 \check{q} 从 Φ 中剔除, 令 $t = t_{\check{q}}, l = l_{\check{q}}$ 。执行 Step2。

其中, Step10、Step11、Step12 是翻箱的启发式规则。候选栈被分为三类, 一类是当前所堆存的集装箱的提箱顺序均晚于待倒箱提箱顺序的栈, 称为第一类候选栈。一类是当前没有集装箱堆存的栈, 称为空栈。另一类是当前所堆存的集装箱中至少有一个集装箱的提箱顺序早于待倒箱提箱顺序的栈, 称为第二类候选栈。将阻碍箱移到第一类候选栈与空栈均不会引起二次翻箱, 而将阻碍箱移到第二类候选栈会引起二次翻箱。因此, 落箱位选择的优先顺序为: 第一类候选栈 \rightarrow 空栈 \rightarrow 第二类候选栈。

$\hat{g}(i, j)$ 是由贪婪算法得到的贝位 (i, j) 范围内所有目标箱的提箱总成本之和, $\hat{g}(k, r)$ 是 $\hat{g}(r, b_{k+1}-1), \hat{g}(b_{k+1}, b_{k+2}-1), \dots, \hat{g}(b_m, \alpha)$ 的最小和。

用 $\hat{g}(\cdot, \cdot)$ 和 $\hat{f}(\cdot, \cdot)$ 替换式 (31) 和 (32) 中的 $g(\cdot, \cdot)$ 和 $f(\cdot, \cdot)$, 得到式 (33) 和 (34)。

$$\hat{f}(m, r) = \hat{g}(r, \alpha), \quad r = m, m+1, \dots, \alpha \quad (33)$$

$$\hat{f}(k, r) = \min_{j=r+1, r+2, \dots, \alpha-m+k+1} \{\hat{f}(k+1, j) + \hat{g}(r, j-1)\},$$

$$k \leq r \leq \alpha - m + k, 1 \leq k < m \quad (34)$$

递归地使用式 (33) 和 (34) 可以求出所有目标箱提箱总成本的最小和 $\hat{f}(1, 1)$, 同时确定每台龙门吊的作业范围, 目标箱的提箱顺序以及翻倒箱的落箱位。

4 算例分析

选择深圳赤湾集装箱码头进行算例分析。一个贝位编号从 01 到 79 的街区被分配两台龙门吊,

待提进口箱随机分布于堆场几个贝位。在进口箱堆存区域的利用率为 50%、60%、70% 以及 80% 的情况下,分别随机产生待提进口箱的初始堆存状态,对这 4 个算例进行测试。

堆场每个贝有 6 个栈,每个栈的最大堆存高度为 5 层。龙门吊大车移动速度为 0.5m/s,移动时加速和减速所用时间之和为 40s,翻箱时间平均为 2min/TEU,龙门吊移动成本为 1 元/贝,翻箱成本为 20 元/TEU,外集卡等待成本为 0.2 元/min。外集卡到港时间服从平均时间间隔为 3min 的负指数分布,预计离港时间为到港之后 10min,最晚预计离港时间为到港之后 30min,计划期为 [0, 50h]。运用 Microsoft Visual C + + 6.0 开发算法的程序,所得结果均在 Inter(R) Core(TM) i5 - 4570 3.20 GHz CPU 及 4 GB RAM 的 PC 上测得。

4.1 模型求解结果

在进口箱堆存区域的利用率为 50% 的情况下,随机产生的待提进口箱的初始堆存状态如图 3

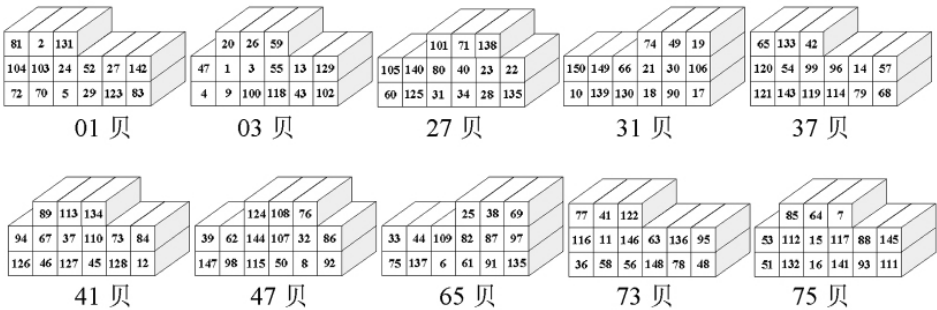


图 3 待提进口箱的初始堆存状态(进口箱堆存区域利用率为 50%)

表 1 堆场翻箱与外集卡提箱顺序同步优化模型求解结果

算例			求解结果					
编号	堆存利用率	箱量 (TEU)	总成本 (元)	翻箱率	翻箱成本 (元)	龙门吊移动成本(元)	龙门吊任务分配	计算时间(s)
1	50%	150	1902	38.0%	1140	612	1: (01 - 47), 2: (49 - 79)	1.13
2	60%	180	2470	40.6%	1460	825	1: (01 - 37), 2: (39 - 79)	1.71
3	70%	210	3467	54.8%	2300	912	1: (01 - 37), 2: (39 - 79)	2.45
4	80%	240	4727	66.7%	3200	1149	1: (01 - 41), 2: (43 - 79)	3.39

由表 1 首先可以看出,总成本、翻箱率、翻箱成本以及龙门吊移动成本这几项指标的值均随着进口箱堆存密度的增大而增大,这符合实际情况。同时,计算时间也随着进口箱堆存密度的增大而增加,这是因为随着目标箱数量的增多,程序运行需要更长的时间。其次,龙门吊的任务分配不同,这是因为不同算例对应不同的优化结果。

4.2 不同提箱方式的比较

为验证堆场翻箱与外集卡提箱顺序同步优化的效果,将其求解结果与其他三种提箱方式的结果进行对比。目前,国内大多数集装箱码头对外集卡

所示,集装箱上的数字为其外集卡的到港顺序,待提进口箱集中堆存在 01 贝、03 贝、27 贝、31 贝、37 贝、41 贝、47 贝、65 贝、73 贝、75 贝,以下称这些贝位为实际贝位。在进口箱堆存区域的利用率为 60%、70% 以及 80% 的算例中,待提进口箱同样集中堆存在以上贝位。

用上述算法对 4 个算例分别进行测试,求解结果如表 1 所示。选取总成本、翻箱率、翻箱成本、龙门吊移动成本、龙门吊任务分配以及计算时间对结果进行比较与分析。对于算例 1 的求解结果,龙门吊 1 的工作范围为(01 贝,47 贝),其中实际贝位为 01 贝、03 贝、27 贝、31 贝、37 贝、41 贝和 47 贝,共 105 个待提进口箱,其中有 32 个目标箱的提箱顺序被调整;龙门吊 2 的工作范围为(49 贝,79 贝),其中实际贝位为 65 贝、73 贝和 75 贝,共 45 个待提进口箱,其中有 13 个目标箱的提箱顺序被调整。

提箱采取先到先服务(FCFS) 的服务原则,翻箱时采取就近安全堆放的原则,即选取最近最矮的栈作为翻倒箱的落箱位,为方便表述,将这种方式称为 FCFS&NR(first come first serve, no rehandle optimization),将本文的方式称为 S&R(sequence optimization, rehandle optimization)。另外设计两种提箱作业方式,FCFS&R(first come first serve, rehandle optimization) 和 S&NR(sequence optimization, no rehandle optimization)。FCFS&R 即先到先服务,优化翻箱方案,其求解算法将单台龙门吊调度优化算法中的 step2-step7 替换为按照先到先服务的原则

选取目标箱; S&NR 即优化外集卡提箱顺序, 不优化翻箱方案, 其求解算法将 step10-step12 替换为按照就近原则选取落箱位。

用 5.1 中的 4 个算例对以上三种方式分别进行测试, 求解结果分别如表 2、表 3 和表 4 所示, 其

各项指标的变化趋势与原因同表 1。将四种提箱方式下的总成本、翻箱率、龙门吊移动成本和计算时间进行对比, 对比结果分别如图 4、图 5、图 6 和图 7 所示。

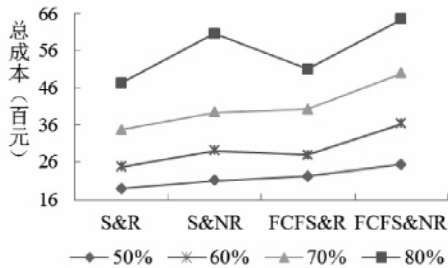


图 4 总成本对比

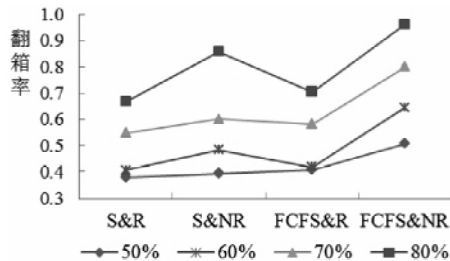


图 5 翻箱率对比

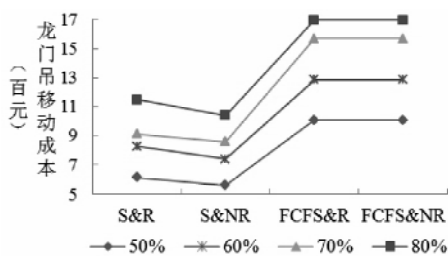


图 6 龙门吊移动成本对比

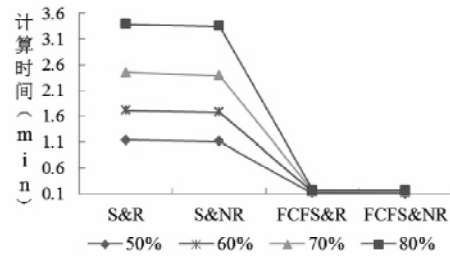


图 7 计算时间对比

表 2 FCFS&NR 方式求解结果

算例			求解结果					
编号	堆存利用率	箱量 (TEU)	总成本 (元)	翻箱率	翻箱成本 (元)	龙门吊移动成本(元)	龙门吊任务分配	计算时间(s)
1	50%	150	2529	50.7%	1520	1008	1: (01-47), 2: (49-79)	0.10
2	60%	180	3628	64.4%	2320	1284	1: (01-47), 2: (49-79)	0.11
3	70%	210	4989	80.0%	3360	1570	1: (01-41), 2: (43-79)	0.14
4	80%	240	6451	96.3%	4620	1697	1: (01-41), 2: (43-79)	0.17

表 3 FCFS&R 方式求解结果

算例			求解结果					
编号	堆存利用率	箱量 (TEU)	总成本 (元)	翻箱率	翻箱成本 (元)	龙门吊移动成本(元)	龙门吊任务分配	计算时间(s)
1	50%	150	2228	40.7%	1220	1008	1: (01-47), 2: (49-79)	0.11
2	60%	180	2787	41.7%	1500	1284	1: (01-47), 2: (49-79)	0.13
3	70%	210	4021	58.1%	2440	1570	1: (01-41), 2: (43-79)	0.15
4	80%	240	5108	70.4%	3380	1697	1: (01-41), 2: (43-79)	0.17

表 4 S&NR 方式求解结果

算例			求解结果					
编号	堆存利用率	箱量 (TEU)	总成本 (元)	翻箱率	翻箱成本 (元)	龙门吊移动成本(元)	龙门吊任务分配	计算时间(s)
1	50%	150	2112	39.3%	1180	558	1: (01-27), 2: (29-79)	1.11
2	60%	180	2900	48.3%	1740	737	1: (01-41), 2: (43-79)	1.67
3	70%	210	3925	60.0%	2520	858	1: (01-37), 2: (39-79)	2.39
4	80%	240	6063	85.8%	4120	1038	1: (01-31), 2: (33-79)	3.35

针对总成本,由图4可以看出,S&R的总成本最低,FCFS&NR总成本最高。对比S&R与S&NR可以看出,S&R由于优化了翻箱方案,总成本比S&NR低。同理,FCFS&R总成本比FCFS&NR低。对比S&R与FCFS&R可以看出,S&R由于优化了提箱顺序,总成本比FCFS&R低。同样,S&NR总成本比FCFS&NR低。因此可以得出,单独优化翻箱与提箱顺序均能减少提箱成本,同时优化翻箱与提箱顺序可以进一步降低提箱总成本。

不同提箱方式下的翻箱率比较如图5所示,可以看出,S&R的翻箱率最低,FCFS&NR的翻箱率最高。对比S&R与S&NR可以看出,S&R由于优化了翻箱方案,翻箱率比S&NR低。同理,FCFS&R翻箱率比FCFS&NR低。对比S&R与FCFS&R可以看出,S&R翻箱率比FCFS&R低,这是因为S&R优化了提箱顺序,减少了翻箱量。同理,S&NR翻箱率比FCFS&NR低。因此,优化提箱顺序可以降低翻箱率,同时优化提箱顺序与翻箱方案可以使翻箱率最低。

不同提箱方式下的龙门吊移动成本比较如图6所示,可以看出,S&NR的龙门吊移动成本最低,FCFS&R与FCFS&NR的龙门吊移动成本最高。S&R的龙门吊移动成本比S&NR高,主要是因为S&R优化了翻箱方案,减少了压箱现象。与S&NR相比,S&R方式下有更多的目标箱可以按照其外集卡到港顺序被提取,而龙门吊的移动次数与距离增加,导致移动成本增加。FCFS&R与FCFS&NR的龙门吊移动成本高是因为这两种方式下外集卡均按照其到港先后顺序被服务,龙门吊移动的次数与距离较S&R与S&NR多。因此,优化外集卡提箱顺序可以降低龙门吊的移动成本,但同时优化翻箱会增加龙门吊的移动成本。

图7为不同提箱方式下的计算时间,可以看出,S&R与S&NR的计算时间比FCFS&R与FCFS&NR的计算时间长,这是因为在S&R与S&NR方式下,程序在选择下一个被服务的外集卡的过程中需要遍历每一个目标箱,消耗时间较长。S&R的计算时间略长于S&NR,这是因为在S&R方式下,落箱位有三种,程序在搜索落箱位时需要执行多次判断,而在S&NR方式下,直接选取落箱位。同理,FCFS&R的计算时间略长于FCFS&NR。

综上可以得出,S&R方式下的提箱总成本与翻箱率最低。在龙门吊移动成本为1元/贝,翻箱成本为20元/TEU,外集卡等待成本为0.2元/min时,与FCFS&NR相比,S&R方式下堆场翻箱率降

低31.06%,提箱作业总成本节省28.49%;与FCFS&R相比,S&R方式下堆场翻箱率降低3.70%,提箱作业总成本节省11.81%。

5 结论

本文针对集装箱码头提箱作业环节,以提箱总成本最小为目标建立了堆场翻箱与外集卡提箱顺序同步优化模型,同时优化外集卡的提箱顺序、龙门吊的任务分配以及堆场翻箱方案。为求解模型,设计了基于动态规划的启发式算法。算例分析表明:与目前码头普遍采用的提箱方式相比,通过调整外集卡提箱顺序并同时优化翻箱方案以及龙门吊的任务分配可以降低堆场翻箱率,减少龙门吊的移动成本,从而节省提箱作业的总成本。

本文模型考虑了码头的作业成本与外集卡的等待成本,有助于均衡各方的利益,提高堆场调度方案的科学性,为集装箱码头降低翻箱率、提高作业效率提供了新思路。模型假设外集卡的到港时间已知,然而,目前码头难以获得外集卡的准确到港时间,因此,部分集卡到达信息情况下如何优化翻箱落位以及外集卡的提箱顺序是值得进一步研究的问题。

参考文献:

- [1] 靳志宏,毛钧,李娜. 基于混合动态规划的集装箱堆场贝位内翻箱作业调度优化[J]. 交通运输系统工程与信息,2011,11(6):131-136.
- [2] Giuliano G, O'Brien T. Reducing port-related truck emissions: the terminal gate appointment system at the Ports of Los Angeles and Long Beach[J]. Transportation Research Part D, 2007, 12(7): 460-473.
- [3] Zehendner E, Feillet D. Benefits of a truck appointment system on the service quality of inland transport modes at a multimodal container terminal[J]. European Journal of Operational Research, 2014, 235(2): 461-469.
- [4] Namboothirya R, Erera A L. Planning local container drayage operations given a port access appointment system[J]. Transportation Research Part E, 2008, 44(2): 185-202.
- [5] 曾庆成,张笑菊,陈文浩,朱晓聪. 基于BCMP排队网络的码头集卡预约优化模型[J]. 系统工程学报, 2013,28(5):592-599.
- [6] Zhao W, Goodchild A V. The impact of truck arrival information on container terminal rehandling[J]. Transportation Research Part E, 2010, 46(3): 327-343.
- [7] 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.

- [8] Kim K H, Hong G P. A heuristic rule for relocating blocks [J]. *Computers & Operations Research*, 2006, 33 (4): 940-954.
- [9] 徐亚,陈秋,龙磊,杨立志,刘丽芸. 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): 3666-3674.
- [10] 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.
- [11] Lee Y, Hsu N Y. An optimization model for the container pre-marshalling problem [J]. *Computers & Operations Research*, 2007, 34(11): 3295-3313.
- [12] Wan Y, Liu J, Tsai P C. The assignment of storage locations to containers for a container stack [J]. *Naval Research Logistics*, 2009, 56(8): 699-713.
- [13] Tang L, Jiang W, Liu J, Dong Y. Research into container reshuffling and stacking problems in container terminal yards [J]. *IIE Transactions*, 2015, 47(7): 751-766.
- [14] Caserta M, VoB S, Sniedovich M. Applying the corridor method to a blocks relocation problem [J]. *OR Spectrum*, 2011, 33(4): 915-929.
- [15] Ünlüyurt T, Aydın C. Improved rehandling strategies for the container retrieval process [J]. *Journal of Advanced Transportation*. 2012, 46(4): 378-393.
- [16] Lee Y, Lee Y J. A heuristic for retrieving containers from a yard [J]. *Computers & Operations Research*, 2010, 37(6): 1139-1147.
- [17] 边展,李向军,靳志宏. 基于规则模拟的堆场取箱作业调度 [J]. *计算机集成制造系统*, 2013, 19(10): 2615-2624.
- [18] Kim K H, Lee K M, Hwang H. Sequencing delivery and receiving operations for yard cranes in port container terminals [J]. *International Journal of Production Economics*, 2003, 84(3): 283-292.
- [19] Ng W C, Mak K L. Yard crane scheduling in port container terminals [J]. *Applied Mathematical Modelling*, 2005, 29(3): 263-276.
- [20] Ng W C. Crane scheduling in container yards with inter-crane interference [J]. *European Journal of Operational Research*, 2005, 164(1): 64-78.
- [21] Li W, Wu Y, Petering M E H, Goh M, De Souza R. Discrete time model and algorithms for container yard crane scheduling [J]. *European Journal of Operational Research*, 2009, 198(1): 165-172.
- [22] Li W, Goh M, Wu Y, Petering M E H, De Souza R, Wu Y C. A continuous time model for multiple yard crane scheduling with last minute job arrivals [J]. *International Journal of Production Economics*, 2012, 136(2): 332-343.
- [23] He J, Chang D, Mi W. A hybrid parallel genetic algorithm for yard crane scheduling [J]. *Transportation Research Part E*, 2010, 46(1): 136-155.
- [24] Chang D, Jiang Z, Yan W. Developing a dynamic rolling-horizon decision strategy for yard crane scheduling [J]. *Advanced Engineering Informatics*, 2011, 25(3): 485-494.
- [25] He J, Huang Y, Yan W. Yard crane scheduling in a container terminal for the trade-off between efficiency and energy consumption [J]. *Advanced Engineering Informatics*, 2015, 29(1): 59-75.