

文章编号: 1001-0920(2011)10-1571-06

## 动态环境下集装箱码头堆场出口箱箱位分配建模与算法研究

周鹏飞, 方波

(大连理工大学 水利工程学院, 辽宁 大连 116024)

**摘要:** 箱位分配是集装箱码头运作优化管理的关键问题. 针对出口箱交箱时间的动态不确定性, 提出两级调度策略, 将出口箱位分配问题分解为贝位分配和贝内具体箱位分配, 分别建立规划模型优化龙门吊大车行使距离和翻箱量, 并针对两级调度策略特点分别设计开发基于禁忌搜索的求解算法. 模拟实验表明, 所提出的方法能够适应动态条件变化, 改善调度效果, 并在合理时间内获得满意解.

**关键词:** 集装箱运输; 动态堆场箱位分配; 规划模型; 禁忌搜索算法

中图分类号: U169.65

文献标识码: A

## Modeling and algorithm research on export container slots allocation under dynamic environments

ZHOU Peng-fei, FANG Bo

(School of Hydraulic Engineering, Dalian University of Technology, Dalian 116024, China. Correspondent: ZHOU Peng-fei, E-mail: pfzhou@dlut.edu.cn)

**Abstract:** Export container slots allocation is a key problem of container port operation management. Focusing on the dynamic uncertainty of export container delivery time, a two-stage scheduling strategy is proposed, and the problem is decomposed into bay allocation in yard and slot allocation in bays, whose object is minimizing the distance of gantry crane and the reshuffle. According to characteristics of two-level scheduling strategy, solving algorithm is developed based on Tabu-search. Experiments show that, the proposed method can efficiently improve allocation effect by dynamic uncertain information, and the algorithm can get the satisfied solution in reasonable time.

**Key words:** container transportation; dynamic slot allocation; programming model; tabu-search algorithm

### 1 引言

堆场是集装箱码头组织生产的关键区域, 其资源主要包括起吊设备和堆场箱位空间, 对于这些资源的调度与管理直接影响到集装箱码头整体运行效率.

针对集装箱码头堆场箱位分配问题, 文献[1]考虑了集装箱重量等级, 运用动态规划模型求解最少翻箱次数的堆放方案; [2]对其翻箱量求解公式进行了修正; [3]对箱位分配的随机堆存策略和分类堆存策略进行了仿真试验分析, 从翻箱率、起重操作量和堆场利用率综合指标来看, 分类堆存策略要明显优于随机堆存策略; [4]利用约束逐步逼近方法研究了出口箱箱位分配求解算法; [5]运用模拟退火算法和仿真技术研究了箱重分布确定条件下的出口箱箱位分配

问题; [6]以重压轻箱数最少为目标, 建立了基于整数规划的选位模型, 并设计了混合智能求解算法; [7]以翻箱最少为目标, 用最小生成树和启发式算法对龙门式起重机小车取箱优化模型进行求解; [8]提出考虑不同重量和不同目的港的堆存模型, 并给出了启发式求解算法; [9]建立了考虑集装箱重量的堆存模型并利用搜索技术求解模型; [10]建立了提交箱不确定条件下的箱位分配模糊机会约束模型.

上述分析表明, 针对箱位分配问题的研究不菲, 由于箱位分配时集装箱的进场序列具有动态随机性, 当前研究大多采用滚动调度的方法来处理问题的动态随机性, 但由于滚动调度具有时间周期性, 滚动周期内的调度效果较差. 本文将研究龙门吊作业工艺条

收稿日期: 2010-06-08; 修回日期: 2010-08-29.

基金项目: 教育部博士点基金项目(200801411105); 中央高校科研业务专项基金项目(DUT10JS10); 辽宁省高等学校科研计划项目(2008Z056).

作者简介: 周鹏飞(1977—), 男, 讲师, 博士, 从事港口物流运作优化、物流系统规划等研究; 方波(1984—), 男, 硕士生, 从事港口规划与港口物流优化的研究.

件下集装箱码头堆场具体箱位的优化分配模型和算法,针对交箱时间的动态随机性,提出了集装箱堆场箱位的两级调度策略;针对问题的层次性,将原问题分解为堆场贝位分配和贝位内具体箱位分配,并设计了基于禁忌搜索的求解实际规模问题的算法。

### 2 两级调度策略

码头出口箱交箱基本过程是: 1) 客户交箱前需向码头预约交箱; 2) 堆场调度人员根据预约交箱序列及相关信息制定箱位分配计划, 交箱序列随着交箱预约和交箱过程动态更新变化; 3) 集装箱进场时, 根据分配计划给集装箱指定箱位, 完成交箱。

考虑码头出口箱交箱特点, 本文将箱位分配分为两个阶段, 形成两级调度: I级, 根据预约交箱时间进行箱位预分配; II级, 交箱时根据动态的交箱序列进行动态分配优化。两级调度策略流程如图 1 所示, 基本流程是: 在滚动调度的基础上, 首先根据预约交箱时间和集装箱信息, 为即将进场的集装箱优化分配箱位, 形成初始调度方案。该阶段考虑滚动调度周期内的所有集装箱, 箱量较多, 全局寻优, 精度高。当集装箱进场时, 根据动态交箱信息优化初始调度方案中当前进场箱的箱位, 生成再调度方案; 同时将再调度方案作为下一集装箱进场时再调度的初始调度方案。滚动调度周期结束时, 进行下一阶段的滚动调度。再调度是根据动态条件的变化在初始调度方案基础上的局部寻优, 要求算法求解效率高。

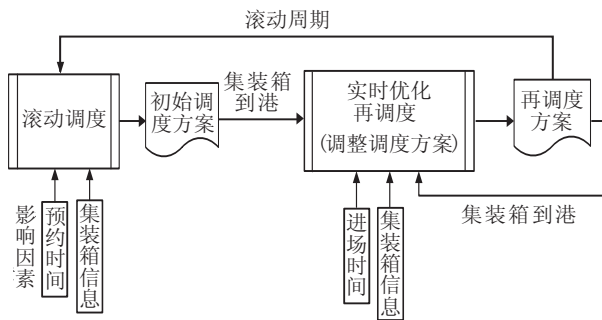


图 1 两级调度策略

### 3 出口箱箱位规划模型

出口箱箱位分配可分解为堆场贝位分配和贝位内具体箱位分配。

#### 3.1 模型假设

- 1) 待分配箱箱区已知, 根据相应船舶、泊位确定。
- 2) 根据交箱预约, 预约交箱时间、重量、目的港等信息已知。
- 3) 实际交箱与预约交箱的时间差近似服从正态分布。
- 4) 贝内需预留翻箱位, 如 4~6 贝位, 需留 3 个翻

箱位。

5) 箱型一致且属于同一船舶。

6) 考虑长途箱和重箱先装船要求, 翻箱量按照短途箱压长途箱和轻箱压重箱的原则计算。

#### 3.2 数学表达

1) 主要符号与变量说明

$K$ : 表示区段中贝位的数量。

$J$ : 表示贝位中堆垛的数量。

$I$ : 表示贝位中堆层的数量。

$N$ : 表示计划进场的集装箱总量。

$M$ : 表示一个充分大的正数。

$d_n$ : 表示第  $n$  个集装箱的目的港。

$w_n$ : 表示第  $n$  个集装箱的重量等级,  $W$  是最大重量等级。

$c_n$ : 表示第  $n$  个集装箱的属性值, 令

$$c_n = W \times d_n + w_n.$$

$V_k$ : 表示第  $k$  个贝位内的容量。

$slot(k, j, i)$ : 表示贝位  $k$  堆垛  $j$  中第  $i$  层的箱位。

$S_{nkji}$ : 表示贝位  $k$  内堆垛  $j$  中第  $i$  层的箱位是否被第  $n$  个集装箱占据, 分别用 0 或 1 表示, 即

$$S_{nkji} = \begin{cases} 1, & \text{第 } n \text{ 个集装箱被分配到 } slot(k, j, i); \\ 0, & \text{otherwise.} \end{cases}$$

$$k = 1, 2, \dots, K, j = 1, 2, \dots, J,$$

$$i = 1, 2, \dots, I, n = 1, 2, \dots, N.$$

$R_{kj(i-z)}$ : 表示堆放在  $slot(k, j, i)$  与  $slot(k, j, i-z)$  上的集装箱是否有翻箱, 即

$$R_{kj(i-z)} = \begin{cases} 1, & \text{集装箱 } a \text{ 堆放在 } slot(k, j, i) \text{ 上, 且对于} \\ & \text{slot}(k, j, i-z) \text{ 的集装箱 } b \text{ 有 } c_a < c_b; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$k = 1, 2, \dots, K, j = 1, 2, \dots, J,$$

$$i = 1, 2, \dots, I, z = 1, 2, \dots, i-1.$$

2) 贝位分配模型

堆场贝位分配目标是龙门吊大车总行驶距离最短和区段内总翻箱量最小。目标函数为

$$\begin{cases} P_1 = \min \sum_{n=1}^N \left| \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{nkji} - \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{(n-1)kji} \right|, \\ P_2 = \min \sum_{k=1}^K \sum_{j=1}^J \sum_{i=2}^I \sum_{z=1}^{i-1} R_{kj(i-z)}. \end{cases} \quad (2)$$

目标函数是带有绝对值的特殊非线性形式, 利用文献 [10] 的方法, 引入两个新的变量  $u_n$  和  $w_n$ , 令

$$u_n + w_n = \left| \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{nkji} - \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{(n-1)kji} \right|,$$

则将目标  $P_1$  转化为线性函数

$$P_1 = \min \sum_{n=1}^N (u_n + w_n). \quad (3)$$

约束条件为

$$\sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I S_{nkji} = 1, \quad n = 1, 2, \dots, N; \quad (4)$$

$$\sum_{n=1}^N S_{nkji} \leq 1,$$

$$k = 1, 2, \dots, K, \quad j = 1, 2, \dots, J, \quad i = 1, 2, \dots, I; \quad (5)$$

$$\sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{0kji} = 1; \quad (6)$$

$$\sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{nkji} - \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{(n-1)kji} +$$

$$u_n - w_n = 0, \quad n = 1, 2, \dots, N; \quad (7)$$

$$u_n \geq 0, \quad w_n \geq 0, \quad n = 1, 2, \dots, N. \quad (8)$$

模型中  $\sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{nkji}$  表示龙门吊大车在堆

垛第  $n$  个集装箱后停留的贝位  $k$ ,  $\left| \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{nkji} \right.$

$\left. - \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I k \cdot S_{(n-1)kji} \right|$  表示两次堆垛作业之间龙

门吊大车行驶的距离. 约束 (3) 表示一个集装箱只能堆放在一个箱位中. 约束 (4) 表示一个箱位最多只能堆放一个集装箱. 约束 (5) 表示龙门吊大车的初始位置在贝位 1 上. 约束 (6) 和约束 (7) 是满足非线性规划函数转化为线性规划函数的条件, 其证明过程详见文献 [10], 此处不再赘述.  $R_{kj(i-z)}$  的计算则由贝位内具体箱位分配模型解得.

### 3) 贝位内具体箱位分配模型

具体箱位分配目标是 minimized 贝内的翻箱量. 目标函数为

$$P_3 = \min \sum_{j=1}^J \sum_{i=2}^I \sum_{z=1}^{i-1} R_{kj(i-z)}. \quad (9)$$

约束条件为

$$\sum_{j=1}^J \sum_{i=1}^I \sum_{n=1}^N S_{nkji} \leq V_k, \quad k = 1, 2, \dots, K; \quad (10)$$

$$\sum_{n=1}^N S_{nkji} \leq \sum_{n=1}^N S_{nkj(i-1)},$$

$$k = 1, 2, \dots, K, \quad j = 1, 2, \dots, J, \quad i = 2, 3, \dots, I; \quad (11)$$

$$\sum_{n=1}^N n \cdot S_{nkji} + \left(1 - \sum_{n=1}^N S_{nkji}\right) M \geq \sum_{n=1}^N n \cdot S_{nkj(i-1)},$$

$$k = 1, 2, \dots, K, \quad j = 1, 2, \dots, J, \quad i = 2, 3, \dots, I; \quad (12)$$

$$\sum_{n=1}^N c_n \cdot S_{nkji} + \left(1 - \sum_{n=1}^N S_{nkji}\right) M -$$

$$\sum_{n=1}^N c_n \cdot S_{nkj(i-1)} + R_{kj(i-z)} \cdot M \geq 0,$$

$$k = 1, 2, \dots, K, \quad j = 1, 2, \dots, J,$$

$$i = 2, 3, \dots, I, \quad z = 1, 2, \dots, i-1. \quad (13)$$

约束 (9) 保证贝位内集装箱的数量不超过贝位容量. 约束 (10) 表示第 2 层以上的集装箱下方堆放集装箱, 不允许悬空放置. 约束 (11) 保证第 2 层以上的集装箱的进场次序不小于其下方集装箱的进场次序. 可以看出, 对于第 2 层以上任意的一个箱位  $\text{slot}(k, j, i)$ , 如果堆放有集装箱, 则等价于  $\sum_{n=1}^N n \cdot S_{nkji} \geq \sum_{n=1}^N n \cdot S_{nkj(i-1)}$ , 即为约束表达; 如果没有堆放集装

箱, 则等价于  $M \geq \sum_{n=1}^N n \cdot S_{nkj(i-1)}$ , 约束满足条件. 约

束 (12) 表示当箱位  $\text{slot}(k, j, i)$  上堆放的集装箱的属性值小于  $\text{slot}(k, j, i-z)$  上集装箱的属性值时, 记 1 次翻箱, 即  $R_{kj(i-z)} = 1$ ; 同时, 对于第 2 层以上任意的一个箱位  $\text{slot}(k, j, i)$ , 如果堆放有集装箱, 则等价于

$\sum_{n=1}^N c_n \cdot S_{nkji} - \sum_{n=1}^N c_n \cdot S_{nkj(i-1)} + R_{kj(i-z)} M \geq 0$ , 即

为约束表达; 如果没有堆放集装箱, 则等价于  $M -$

$\sum_{n=1}^N c_n \cdot S_{nkj(i-1)} + R_{kj(i-z)} M \geq 0$ , 由于目标函数是

最小化翻箱量,  $R_{kj(i-z)} = 0$ , 约束满足条件.

## 4 求解算法

### 4.1 滚动调度求解算法

考虑到模型规模和精度要求, 滚动调度采用基于禁忌搜索算法进行求解, 其基本思想是根据规则生成模型初始解, 并根据模型目标函数和约束迭代搜索箱位逐步逼近最优解, 在迭代搜索过程中记住以往已搜索过的箱位, 在进一步迭代搜索中尽量避开箱位, 使得搜索途径多样化.

根据模型目标函数形式要求构造线性成本函数为箱位交换的评估函数, 形如  $f(x) = c_1 \times P_1 + c_2 \times P_2$ . 其中:  $c_1$  是龙门吊大车行驶一个贝位距离的成本;  $c_2$  是集装箱翻捣一次的成本. 此外, 为便于描述算法, 定义如下符号:  $x^*$  为集装箱最优堆垛方案;  $f(x^*)$  为最优堆放方案下的成本评估值;  $N(x)$  为邻域移动的

集合,即在允许规则下可进行交换的箱位集合; $s$ 为从 $N(x)$ 选取的某一次邻域移动, $s \in N(x)$ ;  $A(s, x)$ 为破禁水平函数; $k$ 为迭代指标; $T$ 为禁忌表; $\max\text{-iter}$ 为最大迭代数。

算法参数如下:

1) 适配值函数:用 $f(x)$ 表示,其值越小代表对应的交换方案越好。

2) 初始解:根据集装箱到港序列,以及“属性值小的在下,属性值大的在上”和“设备移动最小”的规则依次分配箱位,得到初始解。

3) 解的形式:各箱位对应的集装箱属性值、进场序号、调度周期编号,如 $\text{slot}(1, 1, 1) - 25(10, 1)$ ,表示编号为贝位1堆垛1层1的箱位 $\text{slot}(1, 1, 1)$ 堆放一个属性值25,进场序号10,调度周期编号1的集装箱。

4) 禁忌对象,禁忌表:禁忌对象 $T_{mn} = (\text{slot}(k_m, j_m, i_m), \text{slot}(k_n, j_n, i_n), \text{ts})$ ,  $m = 1, 2, \dots, N, n = 1, 2, \dots, N$ ,表示 $m$ 箱所在箱位和 $n$ 箱所在箱位不能互换,禁忌长度为 $\text{ts}$ 。禁忌表为一个矩阵,如表1所示。表1中,方格中的数字为禁忌长度,\*为相同箱位不可交换。

表 1 禁忌表

	slot(1,1,1)	slot(2,1,1)	slot(3,1,1)
slot(1,1,1)	-*		5
slot(2,1,1)		-	1
slot(3,1,1)	5	1	-

5) 禁忌长度:可根据堆场区段规模设定。

6) 邻域集合:一次箱位交换视为一次邻域移动,可交换箱位组构成邻域集合。邻域的选择要满足以下规则:

① 箱位交换的两箱与各自所在堆垛的上下箱应满足顺序堆放的要求(其中 $s_{\text{slot}}$ 表示集装箱的进场次序):

$$s_{\text{slot}}(k_a, j_a, i_a) > s_{\text{slot}}(k_b, j_b, i_b - 1), \quad (14)$$

$$s_{\text{slot}}(k_a, j_a, i_a) < s_{\text{slot}}(k_b, j_b, i_b + 1), \quad (15)$$

$$s_{\text{slot}}(k_b, j_b, i_b) > s_{\text{slot}}(k_a, j_a, i_a - 1), \quad (16)$$

$$s_{\text{slot}}(k_b, j_b, i_b) < s_{\text{slot}}(k_a, j_a, i_a + 1). \quad (17)$$

若 $i_a = 1$ ,则约束(15)成立;若 $i_b = 1$ ,则约束(13)成立;若 $i_a$ 为最高层或为空时,则约束(16)成立;若 $i_b$ 为最高层或为空时,则约束(14)成立。

② 若某箱和空箱位进行交换,则需计算空箱位所在贝位的箱量,满足 $N_k < V_k$ 的贝位才可进行交换移动,其中 $N_k$ 表示贝位 $k$ 内的总箱量,同时需要进行空箱位检验,确保集装箱不悬空。

③ 不考虑与已进场箱的箱位交换。

7) 藐视准则:如果候选方案的适配值小于当前方案的适配值,但它是被禁忌的,则将它解禁。

8) 终止准则:当适配值为0或者迭代次数达到最大迭代次数时,终止算法,输出调度方案。

算法流程如图2所示,具体步骤如下:

**Step 1:** 给出算法参数,按照上文叙述的规则顺序堆放集装箱,并将当前堆垛情况作为禁忌搜索的初始可行解 $x \in X$ 。初始化各参数 $T = \phi, x^* = x, f(x^*) = M$ ( $M$ 为一个充分大的正数), $A(s, x) = f(x^*), k = 0$ 。

**Step 2:** 若 $f(x) = 0$ ,则终止算法,输出调度方案;否则令 $k = k + 1$ 。

**Step 3:** 若 $k > \max\text{-iter}$ ,则终止算法,输出调度方案。

**Step 4:** 针对每一个集装箱进行交换箱位的筛选,生成可交换箱位集合 $N(x)$ ,若有满足藐视准则描述的新调度方案 $x'$ ,则更新破禁水平函数 $A(s, x) = f(x'), x = x'$ ,转至**Step 6**;若不满足藐视准则,则从中选取非禁忌移动产生的新调度方案 $x''$ 。

**Step 5:** 若 $f(x'') < A(s, x)$ ,则更新破禁水平函数 $A(s, x) = f(x''), x = x''$ 。

**Step 6:** 若 $f(x) < f(x^*)$ ,则更新最优调度方案 $x^* = x$ 。

**Step 7:** 更新禁忌表 $T$ ,转至**Step 2**。

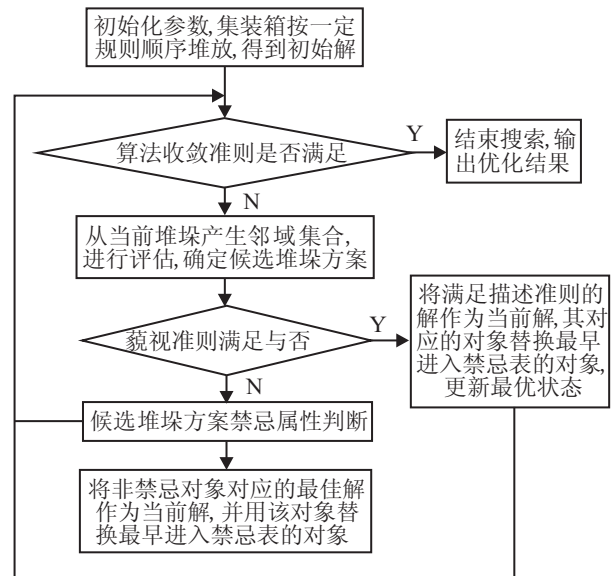


图 2 基于禁忌搜索算法流程

### 4.2 再调度求解算法

为了描述方便,定义符号如下: $S(n)$ 为集装箱可放置域的集合, $f(n)$ 为集装箱可放置域的成本评估函数, $f^*(n)$ 为可放置域最优成本评估值的集合, $P(n)$ 为龙门吊邻域移动的集合, $P_0$ 为龙门吊当前位置, $P_n^*$ 为龙门吊最佳移动位置, $n$ 为集装箱的进场次序,

$N$  为该阶段进场箱总量.

算法流程如图 3 所示, 具体步骤如下:

Step 1: 初始状态与参数初始化,  $P_0$  为龙门吊大车停留位置,  $n = 0, S(n) = \phi, f(n) = \phi, f^*(n) = \phi, P(n) = \phi$ .

Step 2: 若  $n = N$ , 则终止算法, 输出最终堆垛方案; 否则令  $n = n + 1$ .

Step 3: 根据当前堆垛状态生成当前进场箱的可放置域集合  $S(n)$ , 利用选择策略和评估规则对  $S(n)$  进行评估, 记录评估值到  $f(n)$ .

Step 4: 搜索  $f(n)$ , 记录最小值到  $f^*(n)$ , 并记录集合  $f^*(n)$  中的对应贝位到  $P(n)$ .

Step 5: 根据设备移动距离最小原则, 对集合  $P(n)$  进行龙门吊邻域移动评估, 确定最优移动位置  $P_n^*$  及所对应的箱位.

Step 6: 确定箱位, 更新箱位及龙门吊状态,  $P_0 = P_n^*$ , 转至 Step 2.

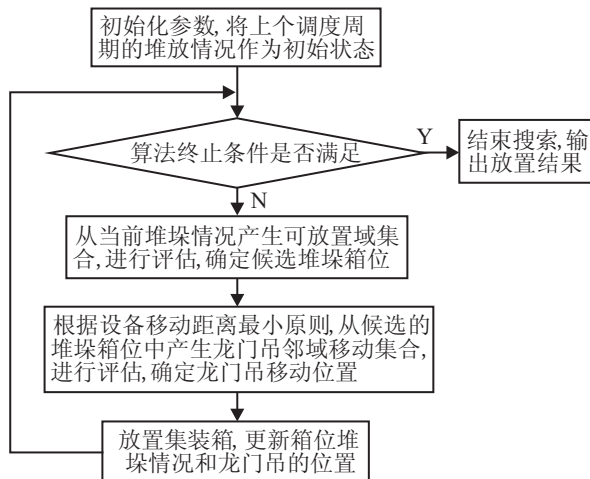


图 3 再调度算法流程图

## 5 仿真实验分析

### 5.1 仿真实验设计

仿真实验取区段规模为 50 个贝位  $\times$  6 个堆垛  $\times$  4 个堆层, 区段内设一台龙门吊, 3 个计算周期, 区段堆存能力为堆场区段容量的 90%. 集装箱的目的港有 3 个. 实验表明: 当前规模下迭代至 150 步后目标值趋于稳定, 因此滚动调度迭代次数取 200 次. 滚动调度周期取 24 h, 周期内进场箱量约为 300 个. 预约交箱时间间隔服从  $\lambda = 13.5$  的负指数分布. 以下实验每组进行 5 次, 结果取均值.

### 5.2 算法性能分析

利用本文算法与 CPLEX 11.2 分别对不同区段规模的问题进行求解, 对比结果见表 2. 表 2 中, \* 为贝  $\times$  栈  $\times$  层; \*\* 为程序运行 40 h 后的解. CPLEX 虽然能够得到模型的最优解, 但只能求解小规模问题,

算法时间消耗随规模呈急速增长态势, 当集装箱量达到 26 时算法时间消耗达到 8.5 h, 难以满足实际需要. 本文算法与最优解比较接近, 但时间消耗远小于 CPLEX, 能够求解实际规模问题.

表 2 本文算法与 CPLEX 对比结果

箱位规模 (箱量)	龙门吊行驶距离		区段内翻箱量		算法消耗时间	
	本文算法	CPLEX	本文算法	CPLEX	本文算法	CPLEX
$1 \times 3 \times 3^*$ (6)	0.0	0.0	0.4	0.2	0.003	0.018
$2 \times 3 \times 3$ (12)	1.2	1.2	0.6	0.0	0.031	0.537
$1 \times 5 \times 4$ (15)	0.0	0.0	1.8	0.6	0.003	1.319
$2 \times 4 \times 4$ (22)	1.4	1.2	0.8	0.4	0.069	3 560.261
$3 \times 4 \times 3$ (26)	2.2	2.0	0.6	0.4	0.325	30 522.841**
$25 \times 6 \times 4$ (478)	59.4	—	2.2	—	38.453	—
$50 \times 6 \times 4$ (953)	149.0	—	8.0	—	247.453	—

两级策略调度与仅采用基于禁忌搜索的滚动调度和步步最优的实时调度性能比较如图 4 和图 5 所示. 图 4 中, 龙门吊大车行驶和翻翘集装箱的成本比例分别取 1:2, 1:1, 2:1 和 4:1 以及极限情况 0:1 和 1:0. 从图 4 可以看出, 两级调度策略能够针对不同的成本比例优化调度方案, 而仅运用滚动调度策略适应性较

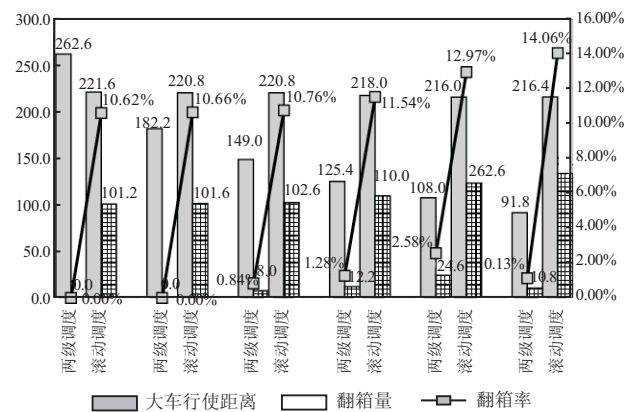


图 4 不同成本比例下的调度效果对比

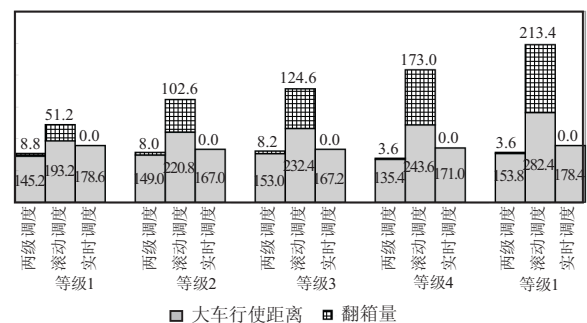


图 5 不同交箱时间动态变化程度下调度效果对比

差,同时两级调度策略总体效果要明显优于仅运用滚动调度的方法.这是由于后者未能按照动态的交箱信息调整调度方案,增加了龙门吊大车行驶距离和区段内的翻箱量.图5显示了实际交箱时间动态变化程度下(5个等级)性能对比.对比结果显示,随着实际交箱时间动态变化程度的增加,仅运用滚动调度的优化效率明显下降,产生大量的翻箱量,龙门吊大车行驶距离也有相应的增加.而两级调度和实时调度算法都相对平稳,同时两级调度策略的优化效果又要明显优于仅运用实时调度的策略.

## 6 结 论

出口集装箱在堆场内的箱位分配问题是一个非线性规划难问题,本文针对交箱序列的动态不确定性提出了箱位分配的两级调度策略;建立了集装箱堆场贝位分配和具体箱位分配模型,在优化翻箱量的同时降低了龙门吊大车行走距离,并设计了基于禁忌搜索的求解算法,实现了预分配与动态优化相结合的调度,改善了调度性能.算法能够满足实际问题规模的求解需要.本文只考虑了单箱型和单船舶的箱位分配问题,针对多箱型以及多船的箱位分配研究将是该领域的重要研究方向.

## 参考文献(References)

- [1] Kim K H, Park Y M, Ryu K R. Deriving decision rules to locate export containers in container yards[J]. *European J of Operational Research*, 2000, 124(1): 89-101.
- [2] Zhang C, Chen W, Shi L, et al. A note on deriving decision rules to locate export containers in container yards[J]. *European J of Operational Research*, 2010, 205(2): 483-485.
- [3] Dekker R, Voogd P, Asperen E. Advanced methods for container stacking[J]. *OR Spectrum*, 2006, 28(4): 563-586.
- [4] Kim K H, Lee J S. Satisfying constraints for locating export containers in port container terminals[C]. *Computational Science and Its Applications-ICCSA 2006*. Berlin: Springer-Verlag, 2006: 564-573.
- [5] Kang J, Ryu K R, Kim K H. Deriving stacking strategies for export containers with uncertain weight information[J]. *J of Intelligent Manufacturing*, 2006, 17(4): 399-410.
- [6] 谢尘,何军良,苍道方.基于混堆模式的集装箱码头出口箱进场选位策略[J]. *上海海事大学学报*, 2008, 29(4): 70-75.  
(Xie C, He J L, Chang D F. Dynamic storage space allocation strategy for outbound containers based on hybrid stack mode[J]. *J of Shanghai Maritime University*, 2008, 29(4): 70-75.)
- [7] 张维英,林焰,纪卓尚,等.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*, 2006, 30(2): 314-317.  
(Zhang W Y, Lin Y, Ji Z S, et al. Optimization model of containers loading operation in export containers terminal[J]. *J of Wuhan University of Technology*, 2006, 30(2): 314-317.)
- [8] 陈庆伟,王继荣.集装箱堆场出口箱堆存模型及其算法[J]. *物流科技*, 2007, 30(7): 106-108.  
(Chen Q W, Wang J R. Modeling and algorithm on piling outbound for container terminal[J]. *Logistics Sci-Tech*, 2007, 30(7): 106-108.)
- [9] 计三有,高悦文.集装箱堆场减少倒箱率方法研究[J]. *水运工程*, 2006(8): 53-56.  
(Ji S Y, Gao Y W. Reduction of rate of container transposition in container yard[J]. *Port & Waterway Engineering*, 2006(8): 53-56.)
- [10] 刘艳,周鹏飞,康海贵.集装箱堆场箱位分配模糊优化研究[J]. *水运工程*, 2009(11): 1-5.  
(Liu Y, Zhou P F, Kang H G. Fuzzy optimization of slot allocation in a container terminal[J]. *Port & Waterway Engineering*, 2009(11): 1-5.)
- [11] 赵银明.一类特殊的非线性规划问题的求解[J]. *江汉大学学报*, 2008, 36(3): 26-27.  
(Zhao Y M. Solution of special nonlinear programming problems[J]. *J of Jiangnan University*, 2008, 36(3): 26-27.)

(上接第1570页)

- [8] Wu L G, Ho D W C. Sliding mode control of singular stochastic hybrid systems[J]. *Automatica*, 2010, 46(4): 779-783.
- [9] Wu L G, Shi P, Gao H J. State estimation and sliding-mode control of Markovian jump singular systems[J]. *IEEE Trans on Automatic Control*, 2010, 55(5): 1213-1219.
- [10] Boukas E K. *Stochastic switching systems: Analysis and design*[M]. Berlin: Birkhäuser, 2005.