

文章编号:1003-207(2018)09-0085-12

DOI:10.16381/j.cnki.issn1003-207x.2018.09.009

考虑实时预倒箱的出口箱堆场多场桥调度优化

郑红星,刘保利,匡海波,闫叙

(大连海事大学交通运输管理学院,辽宁 大连 116026)

摘要:在出口集装箱堆场的实际作业过程中,倒箱是制约场桥作业效率的瓶颈之一。为提高出口箱堆场的作业效能,减少船舶装船作业时间,采用实时预倒来降低倒箱的影响,研究出口箱堆场多场桥调度优化问题。考虑待提箱作业次序固定、场桥间保持安全距离及不可跨越的现实约束,兼顾内集卡的等待上限,侧重场桥作业过程中的实时预倒箱,构建了以带惩罚因子的内集卡总等待时间最少为目标的混合整数线性规划模型。基于问题自身的特点设计了混合和声模拟退火算法,得出了各场桥的行走路径与实时预倒箱方案。在算例实验中,通过与不考虑实时预倒箱的方案、FCFS方案以及下界进行对比,验证了考虑实时预倒箱的场桥调度模型及算法的有效性,为集装箱码头出口箱堆场的场桥调度提供参考。

关键词:实时预倒箱;多场桥调度;出口箱堆场;混合和声模拟退火算法

中图分类号:U693+.7

文献标识码:A

1 引言

由于我国大多数集装箱港口是进口箱和出口箱分开堆放的,在现有岸边作业系统和水平运输系统一定的情况下,出口箱堆场的作业效率直接影响着整个码头的服务水平,研究该类堆场的效率提升策略是当前集装箱港口研究的热点。作为堆场中重要的作业设备—场桥,对其如何科学有效地安排调度来提高装船效率,是出口箱堆场亟需解决的问题。

目前,针对出口箱堆场场桥调度问题,国内外有很多学者进行了深入的研究。鉴于倒箱因素直接影响场桥作业的时长,本文将已有文献按处理该因素的方法不同,分为以下三类:A类——不考虑倒箱影响,只考虑场桥在作业过程中抓取待提箱的时长;B类——固定倒箱次序,在作业过程中场桥抓取某个待提箱的时长等于单独提该箱的时间加上其倒箱时间,倒箱时间或直接给出,或按单箱翻倒时间与压箱数乘积计算;C类——提前控制倒箱,在船舶装船作业开始之前或按配载图提前倒箱,或通过箱位优化确定最少倒箱数的堆存方案。

在A类文献中, Kim等^[1]以出口集装箱作业序列为研究对象,以场桥在各箱区间总移动时间最小为目标,构建了单台场桥作业调度的混合整数规划模型。Mak等^[2]研究了混堆堆场内的多场桥调度问题,考虑了集装箱的提箱准备时间、提箱作业时间、场桥移动时间及因场桥间干扰引起的等待时间,以多台场桥作业完成时刻之和最小为目标,构建了混合整数规划模型,设计了一种遗传算法与禁忌搜索算法相结合的混合优化算法。Chang Daofang等^[3]研究了单个箱区内的双场桥调度问题,建立了以最小化延迟任务数为目标的整数规划模型,并设计了基于滚动时域的启发式算法。乐美龙等^[4]针对出口箱堆场,考虑场桥不可跨越与保持安全距离等现实约束,以多场桥完成所有装卸任务的总时间最短为目标,并设计了两阶段算法,最终给出了一个任务分配和排序方案。Li Wenkai等^[5]研究了单个箱区内的多场桥调度问题,以最小所有集装箱的提箱提前时间、提箱延迟时间及存箱延迟时间的加权和为优化目标,构建了连续型混合整数线性规划模型,并设计了滚动时域算法,实验表明所提出的算法可有效求解大规模算例。Wu Yong等^[6]针对与Li Wenkai等^[5]相同的问题,将提箱延迟任务数引入至目标函数中,提出了一种基于集群—重新分配思想的启发式算法。韩晓龙等^[7]研究了集装箱码头装船作业中的场桥调度问题,结合配载计划,建立了提箱作业总完成时间

收稿日期:2016-07-23; 修订日期:2017-07-13

基金项目:国家自然科学基金资助项目(71473024)

通讯作者简介:郑红星(1971—),男(汉族),河北人,大连海事大学交通运输管理学院,副教授,研究方向:物流系统优化仿真,E-mail:zhredstar@aliyun.com.

最短为目标的场桥调度模型。范厚明等^[8]提出了出口箱箱位分配和场桥调度的分区域平衡策划方法,考虑了场桥间相互干涉的现实约束,对场桥非装卸时间进行了优化,并均衡了各场桥的作业任务量。

在 B 类文献中,Chen Lu 等^[9]研究了出口箱堆场内的多场桥调度问题,同时考虑了场桥间作业干扰、场桥在多个箱区内的转场情况和场桥于单个箱区内的排序情况等,以所有集装箱的最迟提箱完成时刻最小为优化目标,构建了混合整数规划模型,并分别提出了遗传算法和禁忌搜索算法两种启发式算法用于求解。郑红星等^[10]研究了混堆箱区某时段内的多场桥调度问题,考虑了内外集卡的优先级、场桥间不可跨越和保持安全距离等因素,构建了多场桥调度模型,并设计了混合遗传算法用于求解,文中各任务箱的提取操作时间不尽相同。梁承姬等^[11]研究了混堆堆场内某箱区的场桥调度问题,建立了以场桥装卸完工时间最小为目标的优化模型,并设计了相应的优化算法,该文中将各任务箱的提取时间视为已知,但各不相同。虽然上述文献考虑的是混堆,但没有外集卡的装船阶段可看成出口箱堆场。郑红星等^[12]为解决混堆装船箱区内的多场桥调度问题开发了一种滚动时域算法,重点考虑内外集卡的不同优先级别和倒箱因素对场桥调度问题的影响,在减少集卡等待成本的同时降低了倒箱量;且该文中倒箱的处理方法属于按单箱翻倒时间与压数乘积计算。张笑菊等^[13]针对混堆堆场的集装箱码头装船顺序优化问题,考虑船舶一个贝位集装箱的同贝同步装卸作业,以单台场桥在单个箱区内移动时间和翻箱时间最小为目标,构建了出口集装箱装船顺序优化模型并设计了启发式算法,提高了场桥与岸桥的作业效率。Jin Jiangang 等^[14]以场桥作业成本和移动成本之和最小为目标,对场桥空间分配和场桥配置整合优化问题进行了研究,构建了整数线性规划模型,明确了各时段内场桥的工作区域。

在 C 类文献中, Lee 等^[15]将装船前的出口集装箱翻箱问题定义为预翻箱问题,并提出整数规划模型和近邻搜索算法,有效地解决了以翻箱次数最少为目标的堆场翻箱问题。朱明华等^[16]基于给定出口箱堆场集装箱的堆存状态和装船顺序,建立了以倒箱量最少和场桥代价最低为目标的数学模型,提出了定向搜索算法进行求解。周鹏飞等^[17]针对交箱序列的动态不确定性,建立了出口箱堆场中具体箱位分配模型,在优化翻箱量的同时缩短了场桥大车行走距离。邵乾虔等^[18]结合出口箱堆场作业时

间,建立了以最小化预翻箱量和场桥堆存作业移动距离为目标的数学模型,并给出了求解算法。

同时,鉴于出口箱堆场的核心作业是内集卡提箱装船,研究该类堆场的场桥调度需着重考虑内集卡作业效率对其的影响。有关集卡的研究一直是国内外学者研究的热点,其中 Amini 等^[19]为优化集装箱码头集卡拥堵问题,针对集装箱交叉堆放的情况下,构建了多目标线性规划模型并结合三种启发式算法进行求解。张芳芳等^[20]考虑了堆场可存储位置动态变化的因素,以内集卡完成任务的总时间最小为目标,构建了集卡调度模型并采用四种不同的改进 PSO 算法进行求解。赵金楼等^[21]分别以集卡行驶距离和集卡任务间空载距离最小为目标,提出了集卡两阶段路径优化模型,在考虑集卡数量和作业顺序的影响下提高堆场作业效能。

综上,已有关于场桥调度的文献大都以场桥行走路径最短、场桥作业完成时间最短为优化目标,其中忽略倒箱的文献较多,直接将倒箱时间加入场桥作业过程的文献较少,而考虑提前控制倒箱的文献近年来逐渐增加。但在实际作业中限于集港时出口箱抵达时机同配载计划的不匹配,倒箱对场桥作业的时长影响很大,必须在调度过程中加以考虑;同时由于提前倒箱作业场桥数量的不足、场地和时间的限制、装船的时限要求等因素,提前控制倒箱和箱位分配很难实现,因此需考虑在场桥作业过程中的实时预倒箱。在有关集卡调度的文献中,大都以集卡行驶距离最短和总作业时间最短等为目标,少有考虑内集卡等待时间上限和由于场桥倒箱而产生的延误时间,而在出口箱堆场伴有倒箱的装船作业中,为尽可能减少岸桥等待内集卡的时间,需重点考虑如何采用实时预倒箱来加快内集卡的周转。

区别已有文献,本文在现有研究的基础上,针对出口箱堆场的场桥调度优化问题,从降低内集卡等待时间的角度出发,兼顾内集卡提箱的等待容忍限度,重点考虑基于配载计划中待提箱上压箱翻倒的时机和分配,视倒箱为作业过程的另一类别的任务,将倒箱任务和提箱任务集成优化,最终给出优化的场桥作业调度计划,其中包括多台场桥的行走路径和实时预倒箱方案。

2 调度模型及下界

2.1 问题描述

一般而言,预先获知船舶的抵港计划、船期和配载计划后,码头会为其配备合理数量的岸桥,并分配

合适的泊位。从作业重要性上来看,泊位和岸桥是最重要的资源。因此,就出口箱堆场而言,如何高效地进行提箱作业,使得船舶的在泊时间最小,同时实现岸桥利用率的最大化,是出口箱堆场中多场桥调度的核心目的。

本文的问题可描述为:在固定的计划期内,某船舶从出口箱堆场某箱区提箱的配载计划已知,考虑待提箱的提箱次序固定、多台场桥作业过程中不可跨越和保持一定安全距离等现实约束,侧重作业过程中的实时预倒箱,兼顾内集卡的等待上限,使得带有惩罚因子的内集卡总提箱等待时间最少。

2.2 模型假设

本文将待提箱称为主计划箱,将主计划箱上的压箱称为该箱的子计划箱。

模型假设如下:(1)计划期内主计划箱对应内集卡就绪时刻均由往返规律预知;(2)只考虑 20 英尺单一箱型;(3)场桥间不可跨越且须留有安全距离;(4)箱区内主计划箱的箱位分布、配积载图及场桥作业能力等均已知;(5)提箱作业须在对应内集卡就绪后进行;(6)倒箱原则是不压计划期内的主计划箱。

2.3 调度模型

以某箱区为例,其贝位展开图见图 1 示,原问题转化为图论问题见图 2 示。

考虑转化过程中涉及的多主计划箱位于同贝同栈的情况,图 2 右侧即为 Bay4 转化情况(假设 C3 箱计划提箱时刻早于 C5 箱),可见 C3 箱虽为主计划箱,但对其的作业可能不为提箱作业,而是提取其他箱时所需的倒箱作业。

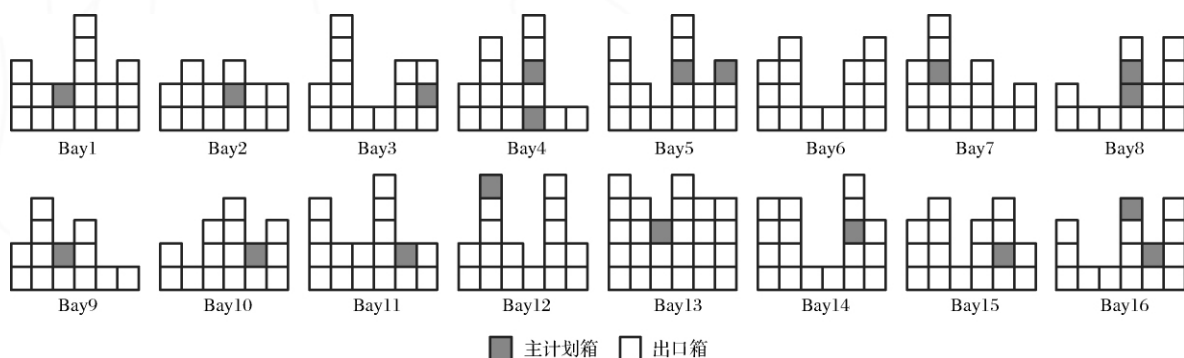


图 1 贝位展开图

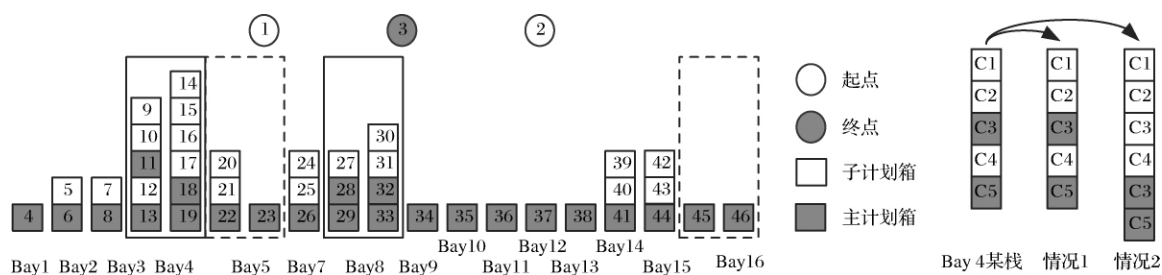


图 2 原调度的图论问题

本文将原调度问题转化为图论问题如图 2 所示,其中实线框内计划箱节点代表原问题中多主计划箱位于同贝同栈的转化情况,虚线框内计划箱节点代表原问题中多主计划箱位于同贝异栈的转化情况(若两种情况都存在则仍用实线框表示);1、2 号节点为起点,代表原问题中场桥;3 号节点为终点,代表原问题中场桥作业完成;4 至 46 号节点为各贝位计划箱转化后所对应的节点。本文调度问题转化为由起点出发,遍历全部中间节点最终汇集至终点

的图论问题;同时在图论问题基础上,引入原问题各箱间作业约束、场桥现实约束等。

参数定义为: M 充分大的正数; QT_i 为 i 箱对应内集卡就绪时刻,不为主计划箱节点时取 M ; D_{ij} 节点 (i, j) 间的距离,以移动时间衡量; B_j 为 j 箱的作业时间,不为箱节点时取 0; C 内集卡惩罚因子; ξ 内集卡等待上限; n 计划箱总数; D 场桥间安全距离; D_i 场桥行走距离 D 所用时长; BW_i 为 i 箱所在贝位,不为箱节点时取 0; Q 实线框外的所有计划箱集合;

R_h 第 h 个实线框内的所有节点集合(节点集合同转化情况相对应); F 所有节点集合; P 场桥节点集合; R 子计划箱节点集合; R' 主计划箱节点集合; FZ 虚拟终点集合; P_k 第 k 号场桥为保持场桥间现实作业约束而无法到达的点集; m 场桥数; L 实线框数; Y_{hv} 第 h 个实线框内第 v 个节点集合; S_{hv} 第 h 个实线框第 v 个节点集合所对应到达边总数; Z_i 若第 i 号节点为主计划箱节点则取 1, 否则取 0。

变量定义为: FT_{ik} 第 k 号场桥作业 i 箱的完成时刻, 若不为箱节点则取非负; X_{ijk} 若第 k 号场桥由节点 i 移至节点 j 则取 1, 否则取 0; A_{ik} 若第 k 号场桥作业 i 箱的等待时间小于 ξ 则取 1, 否则取 0; $C_{ijkk'}$ 若满足 $FT_{ik'} \geq FT_{jk}$ 则取 1, 否则取 0; $E_{ijkk'}$ 若场桥 k' 作业 i 箱同场桥 k 作业 j 箱的作业时间发生冲突则取 1, 否则取 0; $G_{ijkk'}$ 在场桥 k' 作业 i 箱同场桥 k 作业 j 箱作业时间发生冲突的条件下, 若这两个场桥间不满足安全距离约束则取 1, 否则取 0; $P_{ijkk'}$ 在场桥 k' 作业 i 箱同场桥 k 作业 j 箱时作业时间发生冲突且不满足安全距离约束的条件下, 若场桥 k' 先于场桥 k 作业则取 1, 否则取 0; W_{hv} 第 h 个实线框第 v 个节点集合被遍历则取 1, 否则取 0; TT_i 为 i 箱对应内集卡带有惩罚因子的等待时间。

$$\text{Min} Z = \sum_{i \in R'} TT_i \quad (1)$$

$$s. t. : FT_{jk} \geq D_{ij} * X_{ijk} + FT_{ik} - (1 - X_{ijk}) * M + B_j, i, j \in F; k \in P \quad (2)$$

$$FT_{jk} \geq QT_j + B_j - (1 - Z_j) * M - (1 - \sum_{i=1}^{n+m+1} X_{ijk}) * M, j \in F; k \in P \quad (3)$$

$$X_{iik} = 0, i \in F; k \in P \quad (4)$$

$$\sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk} = 1, j \in Q \quad (5)$$

$$\sum_{i=1}^{n+m+1} X_{ijk} = \sum_{i=1}^{n+m+1} X_{jik}, j \in Q; k \in P \quad (6)$$

$$\sum_{k=1}^m \sum_{j=1}^{n+m+1} X_{ijk} = 1, i \in Q \quad (7)$$

$$\sum_{j \in Y_{hv}} \sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk} \leq W_{hv} * M, v \in R_h; h \in L \quad (8)$$

$$W_{hv} \leq M * (\sum_{j \in Y_{hv}} \sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk}), v \in R_h; h \in L \quad (9)$$

$$\sum_{v \in R_h} W_{hv} = 1, h \in L \quad (10)$$

$$\sum_{j \in Y_{hv}} \sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk} = W_{hv} * S_{hv}, v \in R_h; h$$

$$\in L \quad (11)$$

$$\sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk} \leq 1, j \in Y_{hv}; v \in R_h; h \in L \quad (12)$$

$$\sum_{i=1}^{n+m+1} X_{ijk} = \sum_{i=1}^{n+m+1} X_{jik}, j \in Y_{hv}; k \in P; v \in R_h; h \in L \quad (13)$$

$$\sum_{j=1}^{n+m+1} X_{ijk} = 1, i = k; i, k \in P \quad (14)$$

$$\sum_{k=1}^m \sum_{j=1}^{n+m+1} X_{ijk} = 1, i \in P \quad (15)$$

$$\sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk} = 0, j \in P \quad (16)$$

$$\sum_{k=1}^m \sum_{j=1}^{n+m+1} X_{ijk} = 0, i \in FZ \quad (17)$$

$$\sum_{k=1}^m \sum_{i=1}^{n+m+1} X_{ijk} = m, j \in FZ \quad (18)$$

$$FT_{ik} \leq FT_{i+1k'} + (2 - \sum_{j=1}^{n+m+1} (X_{ji+1k'} + X_{jik})) * M, i \in R; k, k' \in P \quad (19)$$

$$FT_{ik} \leq FT_{jk'} + (2 - \sum_{u=1}^{n+m+1} (X_{ujk'} + X_{uik})) * M, QT_i < QT_j; i, j \in R'; k, k' \in P \quad (20)$$

$$FT_{jk} \geq FT_{ik'} - C_{ijkk'} * M, i, j \in F; k, k' \in P \quad (21)$$

$$FT_{ik'} \geq FT_{jk} - (1 - C_{ijkk'}) * M, i, j \in F; k, k' \in P \quad (22)$$

$$FT_{jk} \geq FT_{ik'} + B_j - C_{ijkk'} * M - E_{ijkk'} * M, i, j \in F; k, k' \in P \quad (23)$$

$$FT_{ik'} \geq FT_{jk} - B_j - C_{ijkk'} * M - (1 - E_{ijkk'}) * M, i, j \in F; k, k' \in P \quad (24)$$

$$BW_i \leq BW_j + D + (1 - G_{ijkk'}) * M + (1 - E_{ijkk'}) * M, i, j \in F; k, k' \in P \quad (25)$$

$$BW_i \geq BW_j + D - G_{ijkk'} * M - (1 - E_{ijkk'}) * M, i, j \in F; k, k' \in P \quad (26)$$

$$FT_{jk} \geq FT_{ik'} - B_i + B_j - (1 - P_{ijkk'}) * M - (1 - G_{ijkk'}) * M, i, j \in F; k, k' \in P \quad (27)$$

$$FT_{jk} \leq FT_{ik'} - B_i + B_j + P_{ijkk'} * M + (1 - G_{ijkk'}) * M, i, j \in F; k, k' \in P \quad (28)$$

$$FT_{jk} \geq FT_{ik'} + B_j + D_i - (1 - P_{ijkk'}) * M - (2 - \sum_{u=1}^{n+m+1} (X_{ujk} + X_{uik'})) * M, i, j \in Q, i \neq j; k, k' \in P, k \neq k' \quad (29)$$

$$FT_{ik'} \geq FT_{jk} + B_i + D_i - P_{ijkk'} * M - (2 - \sum_{u=1}^{n+m+1} (X_{ujk} + X_{uik'})) * M, i, j \in Q, i \neq j; k, k' \in P, k$$

$$\neq k' \quad (30)$$

$$X_{ijk} = 0, i \in F; j \in P_k; k \in P \quad (31)$$

$$FT_{ik} - QT_i - \xi \leq (1 - A_{ik}) * M, i \in F; k \in P \quad (32)$$

$$FT_{ik} - QT_i - \xi \geq -A_{ik} * M, i \in F; k \in P \quad (33)$$

$$TT_i \geq FT_{ik} - QT_i - (1 - A_{ik}) * M, i \in F; k \in P \quad (34)$$

$$TT_i \geq (FT_{ik} - QT_i) * C - A_{ik} * M, i \in F; k \in P \quad (35)$$

$$TT_i \geq 0, FT_{ik} \geq 0, i \in F; k \in P \quad (36)$$

式(1)目标函数为带有惩罚因子的内集卡总等待时间;式(2)~(4)保证各场桥顺次作业各计划箱的完成时刻大小关系,且场桥对主计划箱的提箱作业在对应集卡就绪后进行;式(5)~(7)保证实线框外的所有计划箱均被某一场桥作业一次;式(8)~(9)表示变量 W_{hv} 为 1 时实线框 h 中节点集合 v 的到达边大于 1, 否则其到达边为 0;式(10)~(13)保证各实线框内仅有一个节点集合被遍历,且该节点集合内的所有计划箱均被某一场桥作业一次;式(14)~(16)保证各场桥均以对应的场桥节点作为起点,并满足起点的图论约束;式(17)~(18)保证终点

满足图论约束;式(19)保证各栈位的计划箱由场桥从上至下依次作业;式(20)保证主计划箱按集卡到达顺序由场桥依次完成提箱作业;式(21)~(22)表示场桥间作业完成时刻大小同变量 $C_{ijkk'}$ 间的对应关系;式(23)~(24)表示场桥间作业时间是否发生冲突同变量 $E_{ijkk'}$ 间的对应关系;式(25)~(26)表示场桥间是否作业时间发生冲突且不满足安全距离同变量 $G_{ijkk'}$ 间的对应关系;式(27)~(28)表示在不足现实约束的条件下,场桥作业先后次序同变量 $P_{ijkk'}$ 间的对应关系;式(29)~(30)保证各计划箱作业完成时刻的取值;式(31)保证各场桥的作业范围;式(32)~(35)保证不同内集卡等待时间下,带有惩罚因子的等待时间取值;式(36)保证变量范围。

2.4 下界

本文首先给出近似衡量算法有效性的下界 1, 即考虑主计划箱均可在集卡就绪后直接进行提箱作业(不考虑压箱)。下界 1 存在的必要是:无论何种规模均可给出下界。考虑到下界 1 的精确性,本文在原问题基础上松弛场桥间保持安全距离及不可跨越约束,并将同贝同栈情况合理转化,原图论问题转化为新图论问题(下界 2)如图 3 所示。

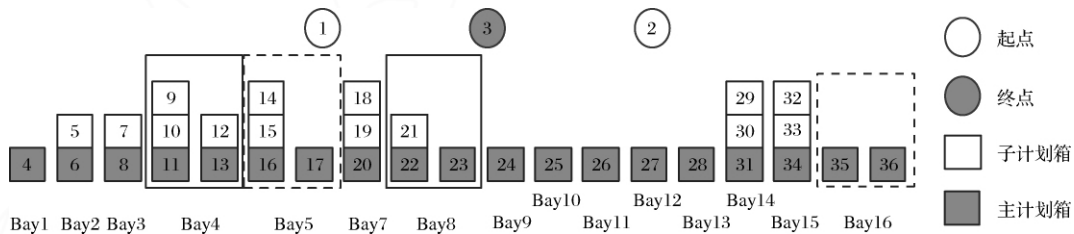


图3 下界 2 图论问题

对本文提出的下界 2 做简要说明。本文在原问题基础上松弛了场桥间保持安全距离及不可跨越的因素,将解空间进行放大,下界 2 解空间是原问题解空间的子集,使得下界 2 最优解必位于原问题最优解之下或与之相同;同时,同贝同栈情况被合理转化,保证转化后作业全部箱所需的总时间变少或不变,因此本文给出下界 2 最优解一定是原问题最优解的下界。

参数定义为: QT_i 为 i 箱对应集卡就绪时刻,不为主计划箱节点时取 M ; D_{ij} 节点 (i, j) 间距离; B_j 为 j 箱作业时间,不为箱节点时取 0; M 充分大正数; C 内集卡惩罚因子; ξ 内集卡等待上限; n 计划箱数; Q 计划箱节点集合; F 所有节点集合; P 场桥节

点集合; R 子计划箱节点集合; m 场桥数; Z_i 若第 i 号节点为主计划箱节点取 1, 否则取 0。

变量定义为: FT_i 第 i 箱完成时刻,若非箱节点则取非负; X_{ij} 由节点 i 移至节点 j 时取 1, 否则取 0; A_{hi} 衡量节点 i 惩罚因子取值, h 取 1 或 2 分别对应两种惩罚情况; TT_i 为 i 箱对应内集卡带有惩罚因子的等待时间。

$$\text{Min} Z = \sum_{i \in R'} TT_i \quad (37)$$

$$s. t. FT_j \geq D_{ij} * X_{ij} + FT_i - (1 - X_{ij}) * M + B_j, i, j \in F \quad (38)$$

$$FT_j \geq QT_j + B_j - (1 - Z_j) * M, j \in F \quad (39)$$

$$FT_k = 0, k \in P \quad (40)$$

$$X_{ii} = 0, i \in F \quad (41)$$

$$\sum_{i=1}^{n+m+1} X_{ij} = 1, j \in Q \quad (42)$$

$$\sum_{j=1}^{n+m+1} X_{ij} = 1, i \in Q \cup P \quad (43)$$

$$\sum_{i=1}^{n+m+1} X_{ij} = 0, j \in P \quad (44)$$

$$\sum_{j=1}^{n+m+1} X_{m+1j} = 0 \quad (45)$$

$$\sum_{i=1}^{n+m+1} X_{m+1} = m \quad (46)$$

$$FT_i \leq FT_{i+1}, i \in R \quad (47)$$

$$FT_i \leq FT_j + (2 - \sum_{u=1}^{n+m+1} (X_{uj} + X_{ui})) * M, i, j \in$$

$$R \text{ and } QT_i < QT_j \quad (48)$$

$$FT_i - QT_i - \xi \leq (1 - A_{1i}) * M, i \in F \quad (49)$$

$$A_{1i} + A_{2i} = 1, i \in F \quad (50)$$

$$TT_i \geq FT_i - QT_i - (1 - A_{1i}) * M, i \in F \quad (51)$$

$$TT_i \geq (FT_i - QT_i) * C - (1 - A_{2i}) * M, i \in F \quad (52)$$

$$TT_i \geq 0, FT_i \geq 0, i \in F \quad (53)$$

式(37)目标函数为带有惩罚因子的内集卡总等待时间;式(38)~(41)保证场桥作业连续性,且内集卡提箱作业在其就绪后进行;式(42)~(44)保证各起点及箱节点的图论约束;式(45)~(46)保证终点的图论约束;式(47)~(48)保证各箱节点间的作业约束;式(49)~(52)保证不同集卡等待时间下,带有惩罚因子的等待时间取值;式(53)保证变量的取值范围。

3 混合和声模拟退火算法

考虑到 CPLEX 仅可在小规模算例时对调度模型进行求解,本文设计了混合和声模拟退火算法(简称 HAS 算法),即以 SA 算法为基本框架,融入 HS 算法中记忆库(HM)及群体操作思想。同进口箱堆场相比,由于船舶配积载图事先已知,出口箱堆场内的主计划箱提箱次序固定,鉴于此本文设计了和声修复策略;同时,为有效提升算法性能,设计了随温度而变化的动态参数。

3.1 HAS 算法流程

本文 HAS 算法流程如图 4 所示, HM 中包括 M 个个体,且每次迭代产生 $M-1$ 个个体,同 HM 中 $M-1$ 个非最优个体按 Metropolis 接受准则取舍,并记录全局最优解。

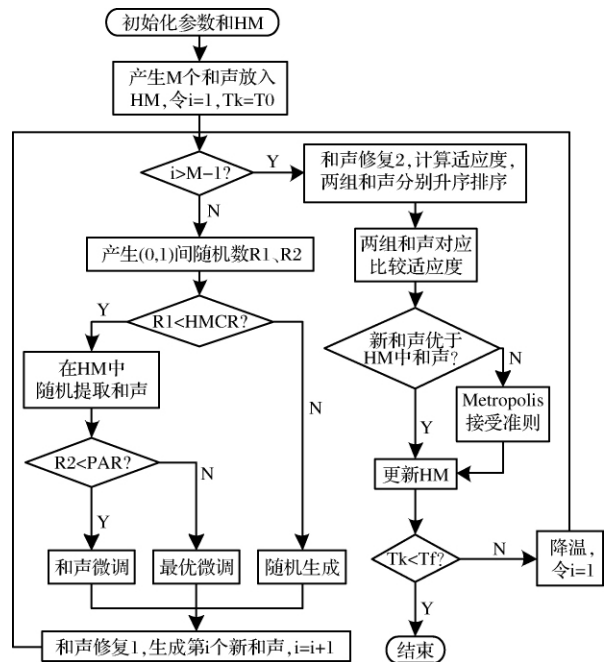


图 4 算法流程图

3.2 解的编码

本文解的编码采用浮点与整数编码结合策略,具体解的编码片段如图 5 所示。其中前段和声为主、子计划箱被分配作业的顺序(各箱按主计划箱及其子计划箱所在层位从上至下依次标号,如 1.1、1.2、1.3 等),后段和声为各箱作业时对应场桥的序号。

以图 5 的某个体片段为例对解的编码做简要说明,首先,本文中每个标号均对应一个集装箱,但一个集装箱可能具有多个标号(如多主计划箱位于同贝同栈)。其次,本文解的编码中各标号(集装箱)的排序按其开始作业顺序先后排序,如图 5 中的 3.1 一定在 1.2 之前作业。最后,前段和声同后段和声长度相同。

3.3 初始化 HM 及随机生成

本文初始化 HM 及随机生成策略相同,为保证产生个体的可行性,设计了产生 M 个个体的步骤如下:

步骤 1 初始化,个体数 $k=1$,主计划箱的次序数 $i=1$;

步骤 2 依次生成第 1 个作业次序 i, j , 其中 $j=1, 2, \dots, G(i)+1, G(i)$ 为 i 箱的压箱数;

步骤 3 令 $i=i+1$, 依次生成第 i 个作业次序 i, j , 其中 $j=1, 2, \dots, G(i)$, 并将第 i 个作业次序随机插入至第 1 个作业次序中;之后,将 i, j 置于第 1 个作业次序的最后,其中 $j=G(i)+1$,

步骤 4 若 i 小于主计划箱数,则转至步骤 3;否

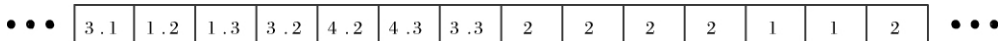


图5 解的编码示意图

则,转至步骤5;

步骤5将更新后的第1个作业次序作为第 k 个个体的前段和声序列;同时,第 k 个个体后段和声的场桥序列随机生成,转至步骤6;

步骤6令 $k=k+1$,若 $k \leq M$,则令 $i=1$,转至步骤2;否则,结束。

3.4 和声微调

本文新个体产生涉及三种方案,分别为和声微调、最优微调(对全局最优和声微调)以及随机生成和声。本文和声微调采用四种单亲变换策略,分别是两点互换策略、正向移位策略、反向移位策略以及逆序变换策略。考虑到前段和声同后段和声的微调策略是分开进行的,本文以某前段和声片段为例对四种策略做简要说明如图6所示。

从图6可以看出和声微调的四种策略均可能使原个体变为不可行,需引入和声修复策略。其次,前、后段和声有某种联系,如各箱因其所在位置不同,而仅可由特定场桥作业,因此仅由前段和声进行四种策略,会降低最优解产生的概率。综上,考虑到迭代初期和声前后段的关联度较低,而在后期和声的前后段关系基本形成。本文在算法后期引入“最优微调”策略,包括上述图6中前三种策略,且“最优微调”均为前、后段和声同时进行变换策略。

3.5 和声修复

针对各主计划箱及其子计划箱间作业顺序约束、各主计划箱间作业顺序约束,本文设计和声修复1;考虑场桥间保持安全距离以及不可跨越因素,本文设计和声修复2(注:在算法中适应度计算部分也会考虑场桥间现实约束,和声修复2起辅助作用)。

3.5.1 和声修复1

考虑到各箱间的作业约束同场桥序号无关,和声修复1仅针对前段和声而进行。以某一新个体为例,具体和声修复1(包括两部分)的步骤如下:

(1)和声修复1的第一部分

步骤1初始化,令 $i=1, j=1$;

步骤2提取个体前段和声第 j 位箱;

步骤3若其为第 i 个主计划箱或其子计划箱,则将第 j 位箱记录到临时信息矩阵中,对应其位置 j 记录到临时位置矩阵中;否则不记录;令 $j=j+1$;

步骤4若 j 大于前段和声长度,则转至步骤5;

否则转至步骤2;

步骤5将临时信息矩阵中的信息升序排序后,依次按临时位置矩阵的原位置替换到原和声中;清空临时信息矩阵、临时位置矩阵;

步骤6令 $i=i+1$,若 i 大于主计划箱数 n ,则结束;否则,令 $j=1$,转至步骤2。

(2)和声修复1的第二部分(对第一部分修复后的个体进行再次修复)

注:临时“前段和声”为零向量。

步骤1初始化,令 $j=L, m=L, k=n$ (其中 L 为前段和声长度, n 为主计划箱数);

步骤2提取个体第 j 位的箱,若该箱是子计划箱,则转至步骤3;否则转至步骤4;

步骤3令临时“前段和声”的第 m 位等于个体前段和声第 j 位的箱, $m=m-1, j=j-1$,若 $j < 1$,则转至步骤7;否则转至步骤2;

步骤4若该箱是第 k 个主计划箱,则令 $k=k-1$ 转至步骤3;否则转至步骤5;

步骤5若 k 大于第 j 位箱对应的主计划箱号,则转至步骤6;否则,令 $j=j-1$,若 $j < 1$,则转至步骤7;若 $j \geq 1$,转至步骤2;

步骤6记录 X 为第 j 位箱所对应的主计划箱序号与 k 的差值,依次产生第 k 个至第 $k-X$ 个主计划箱,并依次放入临时“前段和声”中,令 $k=k-X-1, m=m-X-1, j=j-1$,若 $j < 1$,则转至步骤7;否则,转至步骤2;

步骤7将个体的前段和声更新为临时“前段和声”。

3.5.2 和声修复2

本文的场桥相关约束,需在和声修复中进行有效的限制,具体和声修复2步骤如下:

步骤1初始化,令 $p=1, j=1+L$ (其中 L 为前段和声长度);

步骤2提取第 p 个和声第 j 位场桥序号,记录场桥序号 z ;

步骤3若第 p 个和声第 $j-L$ 位箱所在的贝位,属于第 z 号场桥作业范围,则转到步骤5;否则,转到步骤4;

步骤4将第 p 个和声的第 j 位场桥序号随机更新为可服务此计划箱的场桥序号;



图6 和声微调策略示意

步骤5 令 $j=j+1$, 若 $j>2\times L$, 则转到步骤6; 否则, 转到步骤2;

步骤6 令 $p=p+1$, 若 p 大于和声总个数, 则结束; 否则, 转到步骤2;

3.6 动态参数的设定

本文对 HMCR、PAR 两个参数进行了改进, 引入了随温度而变化的动态参数, 使算法在不同的搜索时期发挥出不同的搜索性能。通过多次实验对比后, 对参数 HMCR 采用反比例曲线 $Y = k/(x+b) + a$, ($k > 0, b > 0$) 进行改进, 使 HMCR 由 0.6 向 0.9 快速收敛, 以增加算法初期的全局搜索能力, 和算法中、后期的局部搜索能力。对参数 PAR 采用 Logistic 曲线: $Y = 1/(K + a * b^x)$, ($a > 0, 0 < b < 1, K > 0$) 进行改进, 使 PAR 值由 0.7 向 0.3 大体上呈 S 型收敛, 保证算法在前期充分进行和声前后段的关联匹配; 同时, 增强了算法后期对全局最优解的改进能力, 提升算法后期的搜索精度。具体的改进见公式(54)~(55)示。

$$HMCR = 179.982 / (T_k + 199.97) \quad (54)$$

$$PAR = \frac{1}{1.428 + 2.1 * 0.00005^{T_k}} \quad (55)$$

3.7 适应度评价

适应度评价见公式(56)~(57)示。

$$QW_i = \begin{cases} 1, & 0 \leq FT_i - QT_i < \xi \\ C, & \xi \leq FT_i - QT_i \end{cases} \quad (56)$$

$$Fitness(X_k) = \sum_{i=1}^n (FT_i - QT_i) * QW_i \quad (57)$$

公式(56)~(57)中, QW_i 为惩罚因子的取值; FT_i 为第 i 个内集卡提箱完成时刻; QT_i 为第 i 个内集卡计划提箱时刻; ξ 为内集卡等待上限; C 为等待时间超过等待上限的惩罚值; X_k 为第 k 个个体; $Fitness(X_k)$ 为第 k 个个体的适应度; n 为主计划箱数。

4 数值实验

4.1 方案有效性实验

(1) 算例原始数据

在出口箱堆场某箱区中, 已知计划期内(据天津港等地的实地调查数据可知, 场桥作业计划约 1h 滚动一次, 故本文取计划期为 1h)内集卡的提箱信息

如表1中原始数据一栏所示。场桥相关信息如下: 跨距6栈, 堆垛高度5层, 移动时间0.1分/贝, 装载任务3分, 倒箱需2分, 作业安全距离3贝。结合港口实地调查数据以及适应度评价公式的特点, 取内集卡等待上限 ξ 为10分、惩罚因子 C 为10。

(2) 算例求解

文中算法的相关参数取值如下: (降温系数, 内循环次数, 初温, 终温, 记忆库大小) = (0.96, 30, 100, 0.01, 5), 所有的实验都运行在 3.10GHz Intel Core 2 CPU 和 4GB 内存的双核计算机上, 采用 MATLAB R2014a 编码。经计算, 本文提出的 HAS 算法搜索过程如图7所示, 搜索到 5800 代左右时, 目标收敛于 45.5 分, 求解耗时 121.29 秒; 场桥实时行走路径如图8所示, 可以看出, 场桥路径无交叉点且始终保持安全距离。

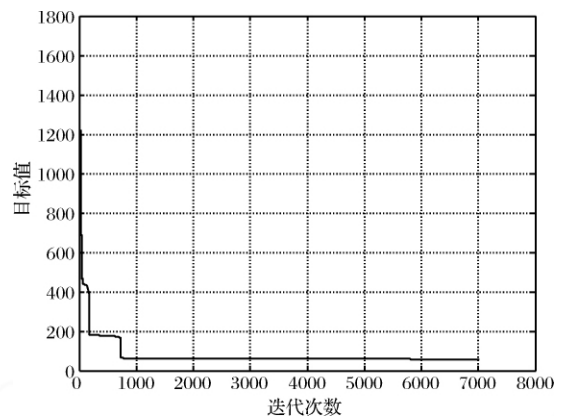


图7 收敛图

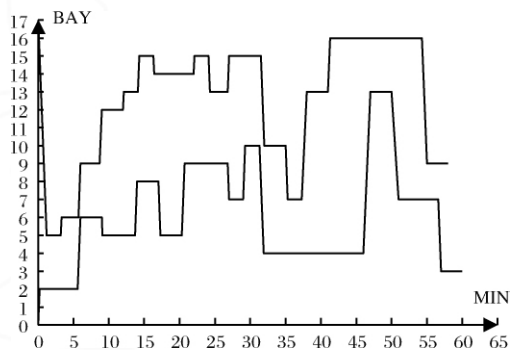


图8 场桥实时行走路径

表 1 方案对比表

任务	原始数据					HAS 方案		RHA 方案		FCFS 方案	
	到达时刻	贝	栈	层	压箱数	完成时刻	等待时长	完成时刻	等待时长	完成时刻	等待时长
1	2	2	4	2	1	5.2	3.2	7	5	7	5
2	6	6	4	1	1	9	3	12.4	6.4	11	5
3	9	12	3	1	0	12	3	12	3	14.6	5.6
4	14	8	4	4	0	17	3	17	3	17	3
5	17	5	4	3	2	20.3	3.3	24	7	24.3	7.3
6	19	14	5	4	1	22	3	24	5	24	5
7	23	9	4	2	1	26	3	29.4	6.4	29.5	6.5
8	27	15	5	2	2	30	3	34	7	37.1	10.1
9	32	10	5	1	1	35	3	37	5	37	5
10	38	13	4	4	1	41	3	43	5	43	5
11	43	4	4	4	1	46	3	48	5	48	5
12	47	13	3	2	1	50	3	52	5	52	5
13	51	7	2	3	2	54	3	58	7	58	7
14	55	9	2	3	1	58	3	63.4	8.4	63.2	8.2
15	57	3	4	2	0	60	3	61.4	4.4	66.8	9.8
目标值	\	\	\	\	\	45.5		82.6		183.4	

为验证本文方法得到场桥作业方案的有效性,分别同采用先到先服务方案和不考虑实时预倒箱的郑红星等^[22]中的方案(在该文中,目标函数基本一致,不涉及内集卡时同本文问题相同)进行对比分析,如表 1 所示。可以看出,FCFS 方案的目标值最大,内集卡作业总等待时间最长,且有一个内集卡超过等待上限;不考虑实时预倒箱的 RHA 方案效果居中,无内集卡超限,较 FCFS 方案好,但是改进有限;本文给出的 HAS 方案效果最好,总等待时间较 RHA 方案降低 44.92%,同下界 1 的相对偏差为 1.11%。

表 2 调度模型的 CPLEX 求解结果

序号	规模	目标值(min)	CPU time(s)
Ins2_5_1	2×5	6	0.53
Ins2_5_2	2×5	6	0.40
Ins2_5_3	2×5	6	0.46
Ins4_5_1	4×5	12	1.95
Ins4_5_2	4×5	12	2.04
Ins4_5_3	4×5	12	1.97
Ins6_5_1	6×5	18	10.95
Ins6_5_2	6×5	—	—
Ins6_5_3	6×5	18	10.68

4.2 算例求解

本文通过 CPLEX Studio IDE 12.5 对原调度模型进行编码求解,随机生成了 30 个算例,考虑小规模下 2、4、6、8、10 个主计划箱和 5 个压箱以及大规模下 12、14、16、18、20 个主计划箱和 10 个压箱(考虑出口箱堆场作业实际,在 1h 内单台场桥最大作业箱量应少于 20)。其中,各主计划箱的位置均随机生成,对应贝号、栈号和层高分别为区间[1,16]、[1,6]和[1,5]内的随机整数;同时,各主计划箱的就绪时刻按经验分布给出,其上的压箱数为[1,4]内的随机整数,得到的结果如表 2 所示。

从表 2 可以看出,CPLEX 的求解时间随着问题规模的增大呈指数型增长,且当规模为 Ins6_5_2 时出现求解时间超过 1 小时的情况。这主要是由于原调度问题出现了多主计划箱位于同贝同栈的情况,由图论转化规则可知,转化后的图论点数呈跳跃式增长,无法在合理的时间内给出最优解,故针对 6 个主计划箱及以上的情况应采用启发式算法进行求解。

4.3 算法的有效性实验

(1)不同规模下的算例比较实验

为验证 HAS 算法的有效性,结合 4.2 中的实际算例,给出算法对比如表 3 所示。针对每一算例,

表 3 算法对比表

序号	总箱数	HAS 算法		下界 1 (min)	下界 2 (min)	下界 1 相对 偏差(%)	下界 2 相对 偏差(%)
		目标值(min)	CPU time(s)				
Ins6_5_1	11	18.00	52.64	18.00	18.00	0.00	0.00
Ins6_5_2	11	18.00	111.47	18.00	18.00	0.00	0.00
Ins6_5_3	11	18.00	53.06	18.00	18.00	0.00	0.00
Ins8_5_1	13	24.00	121.10	24.00	24.00	0.00	0.00
Ins8_5_2	13	27.50	65.57	24.00	27.50	0.15	0.00
Ins8_5_3	13	24.00	63.60	24.00	24.00	0.00	0.00
Ins10_5_1	15	33.50	75.63	30.00	33.50	0.12	0.00
Ins10_5_2	15	33.20	71.92	30.00	32.20	0.11	0.03
Ins10_5_3	15	32.10	110.33	30.00	32.10	0.07	0.00
Ins12_10_1	22	37.60	135.97	36.00	36.00	0.04	0.04
Ins12_10_2	22	37.10	137.00	36.00	36.00	0.03	0.03
Ins12_10_3	22	41.60	124.05	36.00	40.80	0.16	0.02
Ins14_10_1	24	43.20	146.91	42.00	42.00	0.03	0.03
Ins14_10_2	24	48.70	108.16	42.00	45.70	0.16	0.07
Ins14_10_3	24	44.40	107.15	42.00	44.30	0.06	0.00
Ins16_10_1	26	51.10	119.31	48.00	49.50	0.06	0.03
Ins16_10_2	26	53.90	117.60	48.00	48.50	0.12	0.11
Ins16_10_3	26	52.80	120.31	48.00	50.80	0.10	0.04
Ins18_10_1	28	65.30	170.83	54.00	—	0.21	—
Ins18_10_2	28	60.80	169.68	54.00	—	0.13	—
Ins18_10_3	28	55.30	128.93	54.00	—	0.02	—
Ins20_10_1	30	62.20	139.55	60.00	—	0.04	—
Ins20_10_2	30	66.40	137.76	60.00	—	0.11	—
Ins20_10_3	30	66.20	139.99	60.00	—	0.10	—

将本文 HAS 算法目标值同下界进行对比分析。从表 3 中可以看出,在小规模算例下,目标值同下界 1 的平均相对偏差为 4.88%,同下界 2 的平均相对偏差为 0.35%,说明本文提出的 HAS 算法在小规模算例下具有优良的计算效果。在大规模算例下,目标值同下界 1 的平均相对偏差为 9.13%,同下界 2 的平均相对偏差为 4.16%;而在最后 6 个算例下,目标值同下界 1 的平均相对偏差依然控制在 10.1%,说明本文提出的 HAS 算法在大规模算例下依然可以找到近似最优的满意解,验证了算法的有效性。从求解耗时来看,在个别算例中存在突变点,这主要是由于算例中出现了多主计划箱位于同贝同栈的情况,增加了算法的求解耗时,求解耗时整体上呈线性趋势增长,说明算法在不同规模下仍具有较好的稳定性。

(2)不同压箱量的算例实验

针对每组算例,采用本文提出的算法进行求解,对比求解耗时和目标值的增长速度。

从表 4 可以看出,在 15 个算例中,目标函数值和求解耗时都随着压箱量的增加而增加。目标值于压箱量从 20 到 25 时有一个拐点,这是由于在压箱量为 25 时,随着主计划箱及提箱时刻分布的不同,内集卡可用于预倒箱的空闲时间逐渐减少,导致个别内集卡超限。而求解耗时增加较为平缓,表明算法求解时间相对稳定,没有随压箱量的增加而成指数型增长。

5 结语

考虑现实约束,侧重作业过程中的实时预倒箱,本文构建了集装箱码头出口箱堆场多场桥调度的混

表 4 不同压箱量对比表

序号	总箱量	目标值(min)	CPU time(s)	目标值均值	CPU time 均值
Ins15_5_1	20	45.10	96.92		
Ins15_5_2	20	45.00	95.45	46.10	96.60
Ins15_5_3	20	48.20	97.42		
Ins15_10_1	25	46.60	109.44		
Ins15_10_2	25	45.10	109.37	46.33	108.88
Ins15_10_3	25	47.30	107.84		
Ins15_15_1	30	49.20	123.79		
Ins15_15_2	30	45.10	121.29	49.47	121.72
Ins15_15_3	30	54.10	120.09		
Ins15_20_1	35	52.80	136.13		
Ins15_20_2	35	45.10	134.22	53.90	134.84
Ins15_20_3	35	63.80	134.17		
Ins15_25_1	40	50.10	149.50		
Ins15_25_2	40	67.70	147.21	192.70	148.35
Ins15_25_3	40	460.30	148.35		

合整数规划模型,揭示了实时预倒箱对堆场系统作业效率的巨大影响。针对 CPLEX 求解调度模型时存在的问题,本文设计了融入多种改进策略的 HAS 算法对模型进行求解。为验证算法的有效性,文中给出了调度问题的两个下界。数值实验表明,此方法可较好地解决集装箱码头出口箱堆场的多场桥调度问题。

未来研究可扩展为内集卡、场桥和岸桥集成调度中的实时预倒箱问题。

参考文献:

[1] Kim K Y, Kim K H. Heuristic algorithm for routing yard-side equipment for minimizing loading times in container terminals[J]. Naval Research Logistics, 2003, 50(5): 498—514.

[2] Mak K L, Sun D. Scheduling yard cranes in a container terminal using a new genetic approach[J]. Engineering Letters, 2009, 17(4):1—7.

[3] Chang Daofang, Jiang Zuhua, Yan Wei, et al. Developing a dynamic rolling — horizon decision strategy for yard crane scheduling[J]. Advanced Engineering Informatics, 2011, 25(3): 485—494.

[4] 乐美龙,林艳艳,范志强. 基于两阶段启发式算法的多场桥作业调度研究[J]. 武汉理工大学学报,2012,34(1): 60—65.

[5] Li Wenkai, Goh M, Wu Yong, et al. A continuous time model for multiple yardcrane scheduling with last minute job arrivals[J]. International Journal of Production Economics, 2012, 136(2):332—343.

[6] Wu Yong, Li Wenkai, Petering M E H, et. al. Scheduling multiple yard cranes with crane interference and safety distance requirement[J]. Transportation Science, 2015, 49(4): 990—1005.

[7] 韩晓龙,郎昊. 基于模拟退火算法的场桥调度研究[J]. 广西大学学报(自然科学),2015,40(2):427—435.

[8] 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.

[9] Chen Lu, Langevin A. Multiple yard cranes scheduling for loading operations in a container terminal[J]. Engineering Optimization, 2011, 43(11):1205—1221.

[10] 郑红星,于凯. 基于混合遗传算法的混堆箱区内场桥调度研究[J]. 交通运输系统工程与信息,2013,13(5): 150—158.

[11] 梁承姬,周聪,鲁渤. 基于集装箱簇的场桥调度模型与算法[J]. 重庆交通大学学报(自然科学版),2015,35(4):1674—0696.

[12] 郑红星,董译文,匡海波,等. 考虑倒箱的混堆装船箱区内场桥调度优化[J]. 系统工程理论与实践,2016,36(9):2362—2373.

[13] 张笑菊,曾庆成,杨忠振. 基于堆场混合堆存的集装箱码头装船顺序优化模型[J]. 系统工程理论与实践, 2016,36(9):2374—2382.

[14] Jin Jiangang, Lee D — H, Cao Jinxin. Storage yard management in maritime container terminals [J].

- Transportation Science, 2016, 50(4): 1300—1313.
- [15] Lee Y, Hsu N Y. An optimization model for the containerpre-marshalling problem[J]. Computer and Operations Research, 2007, 34(11): 3295—3313.
- [16] 朱明华,程奂翀,范秀敏. 基于定向搜索算法的集装箱堆场翻箱问题[J]. 计算机集成制造系统, 2012, 18(3): 639—644.
- [17] 周鹏飞,李丕安. 集装箱堆场进口箱翻箱落位仿真分析与启发式优选算法[J]. 系统工程理论与实践, 2013, 33(12): 3145—3155.
- [18] 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.
- [19] Amini A, Tavakkoli—Moghaddam R. A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks[J]. Computers & Industrial Engineering, 2016, 96: 180—191.
- [20] 张芳芳,牛奔,李丽,等. 基于堆场可存储位置动态变化的集卡调度智能优化[J]. 中国管理科学, 2016, 24(S1): 217—224.
- [21] 赵金楼,黄金虎,刘馨. 集装箱码头的集卡两阶段路径优化研究[J]. 中国管理科学, 2017, 25(4): 152—157.
- [22] 郑红星,于凯,李芳芳,等. 考虑外集卡的混堆集装箱码头多场桥调度[J]. 计算机集成制造系统, 2014, 20(12): 3161—3169.

Multi-yard Cranes Scheduling Optimization of Export Container Yard Considering Real-time Pre-marshaling

ZHENG Hong-xing, LIU Bao-li, KUANG Hai-bo, YAN Xu

(Transportation and Management College of Dalian Maritime University, Dalian 116026, China)

Abstract: In the actual operation process of the export container yard, pre-marshalling is one of the bottlenecks that constrain the operation efficiency of yard cranes. In order to improve the operation efficiency of the export container yard and reduce the processing time of the shipment, the real-time pre-marshalling is adopted to reduce the impact of marshalling and optimize the scheduling of multi-yard cranes in the export container yard. Due to the fixed operation sequence of containers waiting for lifts and the constraints of non-crossing and safe distance among yard cranes, the real-time pre-marshalling is investigated, as well as considering the waiting time constraints of internal container trucks. A mixed integer programming model with a penalized objective function is proposed to minimize the total waiting time of internal container trucks. Based on the characteristics of the problem, a hybrid harmony simulated annealing algorithm is designed to obtain the walking path of each yard crane and a real-time pre-marshalling scheme. In the numerical experiments, compared with the scheme without considering the real-time pre-marshalling, the FCFS scheme and the lower bound, the effectiveness of the crane scheduling model and the algorithm considering the real-time pre-marshalling is validated, and a reference is provided for the crane scheduling in export container yard.

Key words: real-time pre-marshaling; multi-yard cranes scheduling; export container yard; hybrid harmony simulated annealing algorithm