

干涉存在下自动化集装箱码头穿越式双起重机同步调度优化

王耀宗, 胡志华*

(上海海事大学 物流研究中心, 上海 201306)

摘要: 针对穿越式双起重机作业过程中的干涉问题, 分析并界定干涉的存在形式, 以最小化任务完成时间为目标, 建立穿越式双起重机同步调度下的混合整数规划模型, 确定装卸任务的分配与每台起重机的作业序列. 用状态转移函数描述起重机的作业过程, 提出基于任务优先级的随机键遗传算法, 求解所提出的优化问题. 根据现实自动化集装箱码头的特征参数, 设计不同规模的算例并进行计算实验. 实验结果表明, 模型满足同步调度的约束条件, 所得的解均为无冲突调度, 验证了模型的有效性. 算法求解结果显示, 所得结果能够规避干涉实现同步调度, 验证了算法的合理性与有效性. 此外, 所提算法具有良好的求解性能, 能够提升穿越式双起重机同步调度问题的求解速度.

关键词: 自动化集装箱码头; 双起重机调度; 干涉; 优先权; 遗传算法

中图分类号: F224

文献标识码: A

doi: 10.7511/dllgxb202001012

0 引言

堆场作业是自动化集装箱码头 (automated container terminal, ACT) 的主要环节. 为充分利用堆场空间、提升堆场作业效率, ACT 堆场采用穿越式双起重机 (dual automated stacking cranes, Dual-ASC) 装卸工艺协同作业^[1]. 该装卸工艺的交叉作业特征解除了因作业范围受任务类型的制约, 作业效率的提升体现在两个方面: 不同类型的任务 (存箱与取箱) 可由任意一台 ASC 作业, 避免相互等待; 作业过程中, 两台 ASC 可相互穿越, 解除了因作业范围限制产生的等待时间. Saanen 等^[2] 首先从理论上对穿越式双 ASC 作业可行性进行分析与证明, 之后, 汉堡港 (Hamburger Hafen) 在单箱区 (block) 内配置穿越式双 ASC 作业工艺^[3], 验证了该作业工艺在应用上取得的成效. 然而, 穿越式双 ASC 的调度需要考虑干涉与穿越共同存在, 具有较高的难度^[4]. 为此, 许多学者已开展穿越式双 ASC 调度优化研究, 探索 ASC 的干涉规避策略与穿越规则, 为避免冲突发生、提高作业效率提供有效参考^[5-8].

双 ASC 之间由于干涉的存在导致作业工序

相互耦合以及装卸决策相互依赖^[9], 使 ASC 调度的复杂性显著增加. 国内外相关领域的学者从多个角度展开对双 ASC 调度问题的研究, 研究成果大致可以分为 4 类. 第 1 类: 问题转换与一般性抽象. Vis 等^[10] 把装卸任务抽象为节点, 用点与点连接的弧表示作业时间, 将 ASC 调度问题转化为 TSP; 在二分网络中找到部分问题的最佳匹配, 并通过动态规划连接这些部分求得最优解. Gharehgozli 等^[11] 考虑不同优先级的存取箱任务, 把 ASC 的调度问题抽象为具有优先约束的多重非对称广义 TSP, 以最小化作业完成时间为目标, 设计自适应大邻域搜索算法求解该问题. He 等^[12] 将双 ASC 调度问题转化为带软时间窗的车辆路径问题 (VRP with soft time windows), 以作业延误时间和 ASC 能源消耗最小为目标, 用集成仿真技术搜寻解空间. Vis 等^[3] 把双 ASC 调度问题转化为单 ASC 调度问题, 用分支定界算法求出下界. 第 2 类: 保持两台 ASC 在安全距离范围内运行以规避干涉. Li 等^[13] 考虑在安全距离的约束下, 建立双 ASC 调度优化模型来界定干涉规避的途径. 魏晨等^[14] 考虑双 ASC 协同作业的特征, 以

收稿日期: 2019-06-25; 修回日期: 2019-11-22.

基金项目: 国家自然科学基金资助项目 (71871136); 上海市科学技术委员会重点项目 (17DZ2280200).

作者简介: 王耀宗 (1993-), 男, 硕士生, E-mail: yzwang_hit@163.com; 胡志华* (1977-), 男, 博士, 教授, 博士生导师, E-mail: zhhu@shmtu.edu.cn.

安全距离作为空间约束,优化时空同步与规避干涉导致的相互等待时间. Wu 等^[4]提出了聚类重构算法,根据安全距离的要求调整 ASC 的作业开始时间(或完成时间)以消除干涉.周静娴等^[15]依据任务在同一贝位(bay)的作业情况,建立以安全时间为干涉规避约束的 ASC 调度混合整数规划模型. Hu 等^[16]把安全距离转化为时间间隔,并用矩阵表示任务之间的干涉情况,设计混合整数规划模型刻画批次任务之间的序列约束;针对干涉问题导致的不可行解,设计基于贪心策略的插入算法处理.范厚明等^[17]以安全距离作为相邻场桥间的空间约束,建立单箱区箱位分配及多场桥协同优化模型,并设计遗传算法求解.第 3 类:箱区内设置缓冲区以规避干涉.缓冲区的设置一定程度上解除了两台 ASC 之间的耦合,以进口箱作业为例,具体做法为:海侧 ASC 将集装箱移交(hand over)到缓冲区,之后由岸侧 ASC 接管(take over)该集装箱并运送到目标存储位置. Zhou 等^[18]首先采用该调度策略,考虑在非交叉约束下,通过定义任务优先权的方式协同作业.该策略的关键是制订一个调度计划(时间表),要求所有任务都在其截止日期(due date)之前被移交到缓冲区.但这一约束限制了作业的灵活性,同时增加了另一台 ASC 的等待时间.因此,Briskorn 等^[19]在 Zhou 等^[18]研究的基础上,解除了固定缓冲区的限定,把缓冲区的位置作为决策对象,进而确定最佳的 ASC 调度计划.考虑到缓冲区位置分布对调度结果的影响, Han 等^[20]研究了不同缓冲区位置设定下的 ASC 调度问题.第 4 类:预先定义障碍图(obstacle graph)以规避干涉. Briskorn 等^[21]预先定义干涉发生时 ASC 的规避规则,用分解策略拆分双 ASC 的调度问题并开发强多项式算法求解该问题. Nossack 等^[22]分析了 ASC 作业不同特征任务时对应的状态,提出了一个基于障碍图的穿越式双 ASC 调度整数规划模型.此外, Dorndorf 等^[23]关注穿越式 Triple-ASC(automated triple cross-over stacking cranes)调度,当一个任务完成或一个新任务到达时,动态调整调度计划以优化任务的分配与 ASC 的作业序列.冯媛君等^[24]以堆场翻箱与外集卡提箱顺序同步为目标,建立同步优化模型,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.

上述研究,从问题转化、设置安全距离、引入缓冲区(包括缓冲区位置设置策略),以及预先定

义障碍图等角度展开,提出了相应的求解算法,丰富了穿越式双 ASC 的调度优化理论.本文在现有研究的基础上,分析穿越式双 ASC 作业的特点,界定干涉发生的条件及规避策略,建立穿越式双 ASC 同步调度优化模型,设计基于优先规则的随机键遗传算法求解该问题.

1 问题描述

自动化码头堆场箱区布局如图 1 所示.箱区(block)由若干贝位 l 和列 r 组成,进口箱与出口箱混合堆存,由 ASC 进行存取箱作业.箱区两端各有一个交接区,每个交接区有若干交接位(I/O),集装箱在交接位进出堆场. ASC1 和 ASC2 横跨箱区,分别运行在两条不同的轨道上,可相互穿越,交叉作业存箱任务与取箱任务,作业过程为:(1)当执行存箱作业时,ASC 将集装箱从两端交接位提起并运送到对应的存箱位;(2)当执行取箱作业时,ASC 将集装箱从取箱位提起并运送到两端对应的交接位.当两台 ASC 相互穿越时,ASC1 的吊具要位于 ASC 的远端一侧,为两台 ASC 的穿越提供条件.另外,ASC 作业任务的先后顺序称为 ASC 的作业序列.

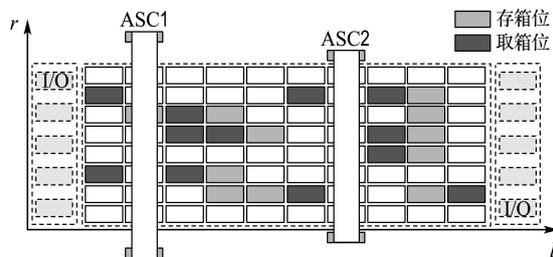


图 1 堆场箱区布局示意图

Fig. 1 Diagram of the yard block's layout

每一任务对应两个位置:起始位置和目标位置.若任务 i 为存箱作业,起始位置 O_i 位于箱区两端交接区,目标位置 D_i 位于箱区贝位;若任务 i 为取箱作业,起始位置 O_i 和目标位置 D_i 分别位于箱区贝位和两端交接区. ASC 作业过程分为 3 个阶段,分别对应一个操作时间:在起始位置抓取并提升集装箱,服务时间 T_1 ;带载运行到目标位置,运行时间 T_{ii} ;在目标位置释放集装箱,服务时间 T_2 .因此,任务 i 的作业时间 T_i 为 3 个阶段操作时间之和,如式(1)所示.在同一贝位上,若其中一台 ASC 执行提升或释放集装箱操作时,另一台 ASC 在该时刻不能执行此类操作.

$$T_i = T_1 + T_{ii} + T_2 \quad (1)$$

在交叉作业的模式下, 每台 ASC 可在整个箱区范围内移动, 不可避免地在同一贝位发生干涉. 因此, 首要任务是界定干涉发生的条件, 为同步调度约束的制订提供依据. 干涉发生的时间关系如图 2 所示. 设 S_i 为任务 i 的作业开始时间, e_i 为任务 i 的作业完成时间. 图 2 界定了 4 种干涉情况: 在同一贝位, 图 2(a) 任务 i 和 j 都为存箱作业, 两者的完成时间相同; 图 2(b) 任务 i 和 j 都为取箱作业, 两者的开始时间相同; 图 2(c) 任务 i 为存箱作业, 任务 j 为取箱作业, 任务 i 的完成时间等于任务 j 的开始时间; 图 2(d) 任务 i 为取箱作业, 任务 j 为存箱作业, 任务 i 的开始时间等于任务 j 的完成时间. 当以上情况出现时, 采用延迟作业的策略规避干涉, 使其中一台 ASC 等待另一台 ASC 执行集装箱提取(或释放)操作完成后, 才允许其继续作业, 等待的时间为 T_w . 因此, 规避干涉的作业时间如式(2)所示. 此外, 选取花费时间成本较小的 ASC 执行等待步骤.

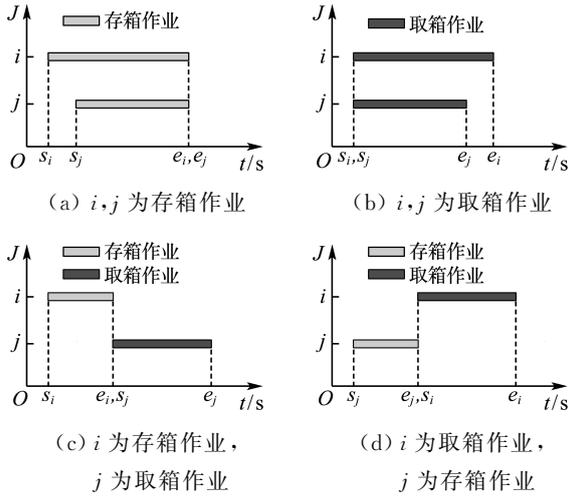


图 2 干涉发生的时间关系
Fig. 2 Time relationship of interference

$$T'_i = T_i + T_w \quad (2)$$

综上, 本文研究单箱区配置跨越式双 ASC 的调度问题, 界定干涉发生的条件与规避处理的方法. 目标是优化任务的分配和每台 ASC 的作业序列, 使任务完成时间最小. 因此, 需要决策每个任务的执行者(ASC)、每台 ASC 执行任务的序列以及是否需要规避干涉.

2 调度优化模型

两台 ASC 协同执行任务: 在 ASC1 作业一个

任务的同时, ASC2 可以作业另一个任务, 作业过程中要规避干涉. 干涉的存在导致作业工序相互耦合以及装卸决策相互依赖, 围绕这种耦合与依赖所产生的基本制约, 称为同步约束. 同步约束界定了干涉发生和规避的处理途径.

为满足 ASC 在调度过程中的同步约束, 当干涉情况出现时, 通过延迟部分任务的开始时间(或完成时间)以规避干涉. 然而, 这将增加部分任务的完成时间, 并影响任务的分配结果与 ASC 的作业序列. 为优化跨越式双 ASC 的作业序列, 最小化任务的完成时间, 建立混合整数规划模型 M. 对于任一任务 i 而言, 需要做以下决策: (1) 任务 i 分配给哪个 ASC 作业; (2) ASC 在任务 i 的起始位置开始作业的时间; (3) ASC 在任务 i 的目标位置结束作业的时间和准备做的下一个任务 j .

已知任务集合 J 、任务数量 n 、类型(存箱或取箱)、贝位集合 B 、任务 i 的起始位置 O_i 和目标位置 D_i . 任务 i 从起始位置到目标位置对应 ASC 的运行时间 T_{ii} 、ASC k 从任务 i 的目标位置 D_i 到任务 j 的起始位置 O_j 的空载时间 T_{kij} 可由已知数据计算得出. 模型中相关的符号定义如下.

集合:

$$J_s, J_r: \text{存箱任务和取箱任务}, J_s \cup J_r = J, J_s \cap J_r = \emptyset.$$

J : 任务集合, 包括存箱任务和取箱任务, 由 i, j 索引.

$$\bar{J}, J^-, J^+: \bar{J} = J \cup \{O, \bar{O}\}, J^- = J \cup \{O\}, J^+ = J \cup \{\bar{O}\}, \text{其中 } O \text{ 和 } \bar{O} \text{ 是两个虚拟任务.}$$

K : ASC 的集合, $K = \{1, 2\}$, 由 k 索引.

B : 贝位集合, $B = \{0, 1, \dots, B_n, B_n + 1\}$, 其中 0 和 $B_n + 1$ 分别是两侧交接区.

参数:

T_b : ASC 移动一个贝位的时间, $T_b = 4 \text{ s}$.

O_i, D_i : 任务 i 的起始位置和目标位置, $O_i, D_i \in B$. O_i, D_i 有一个为箱区两端的交接位置, $J_s = \{i | O_i \in \{0, B_n + 1\}, i \in J\}$, $J_r = \{i | D_i \in \{0, B_n + 1\}, i \in J\}$.

L_i : 任务 i 的目标位置. 如果 $i \in J_r$, 则 $L_i = O_i$; 否则, $L_i = D_i$.

T_{kij} : 任务 i 到任务 j ASC k 的空载时间, $T_{kij} = |O_j - D_i| T_b$, 其中 $i, j \in \bar{J}$, 单位为 s.

T_{ii} : ASC 作业任务 i , 从起始位置 O_i 到目标位置 D_i 的运行时间, $\forall i \in \bar{J}$, 单位为 s.

T_s :ASC 提升或释放一个集装箱的服务时间, $T_s=30$ s.

T_w :规避干涉的等待时间, $T_w=30$ s.

n :任务的数量.

决策变量:

x_{kij} :0-1 变量. $\forall i, j \in \bar{J}$. $x_{kij}=1$, ASC k 依次作业任务 i, j ; 否则, $x_{kij}=0$.

y_{ki} :0-1 变量. $\forall i \in J$. $y_{ki}=1$, 任务 i 由 ASC k 作业; 否则, $y_{ki}=0$.

s_{ki} :连续变量. 任务 i 由 ASC k 作业的开始时间, $\forall i \in \bar{J}, \forall k \in K$.

e_{ki} :连续变量. 任务 i 由 ASC k 作业的完成时间, $\forall i \in \bar{J}, \forall k \in K$.

s_i :连续变量. 任务 i 的开始时间, $\forall i \in \bar{J}$.

e_i :连续变量. 任务 i 的完成时间, $\forall i \in \bar{J}$.

t :连续变量. 所有任务作业完成时间.

a_{ij} :0-1 变量. $a_{ij}=1$, 任务 i 的作业完成时间在任务 j 的作业完成时间之前; 否则, $a_{ij}=0$. 其中 $L_i=L_j, \forall i, j \in J_s$.

b_{ij} :0-1 变量. $b_{ij}=1$, 任务 i 的作业开始时间在任务 j 的作业开始时间之前; 否则, $b_{ij}=0$. 其中 $L_i=L_j, \forall i, j \in J_r$.

c_{ij} :0-1 变量. $c_{ij}=1$, 任务 i 的作业完成时间在任务 j 的作业开始时间之前; 否则, $c_{ij}=0$. 其中 $L_i=L_j, i \in J_s, j \in J_r$.

d_{ij} :0-1 变量. $d_{ij}=1$, 任务 i 的作业开始时间在任务 j 的作业完成时间之前; 否则, $d_{ij}=0$. 其中 $L_i=L_j, i \in J_r, j \in J_s$.

对模型做以下假设:(1)ASC 在轨道上移动的速度及吊具移动速度恒定;(2)当 ASC1 正在执行提升或释放集装箱操作, 而 ASC2 要穿越 ASC1 时, 忽略 ASC1 移动到轨道侧端的时间;(3)箱区两端的交接区容量充足, ASC 在交接区不必等待.

考虑干涉存在下的穿越式双 ASC 同步调度问题, 建立混合整数规划模型 M, 确定每个任务的分配结果和每台 ASC 的作业序列, 使任务完成时间最小. 数学模型表示为

$$M \quad \min f = \max_i e_i \quad (3)$$

$$\text{s. t.} \quad \sum_{j \in J} x_{kOj} = \sum_{i \in J} x_{ki\bar{O}} = 1, \forall k \in K \quad (4)$$

$$\sum_{k \in K} y_{ki} = 1, \forall i \in J \quad (5)$$

$$\sum_{j \in J^+} x_{kij} = y_{ki}, \forall k \in K, \forall i \in J \quad (6)$$

$$\sum_{i \in J^-} x_{kij} = \sum_{i \in J^+} x_{kji}, \forall k \in K, \forall j \in J \quad (7)$$

$$e_{ki} \geq s_{ki} + T_{ii} + 2T_s + (y_{ki} - 1)M, \forall i \in J, \forall k \in K \quad (8)$$

$$s_{kj} \geq e_{ki} + T_{kij} + (x_{kij} - 1)M, \forall i, j \in J, \forall k \in K \quad (9)$$

$$s_{ki} \leq y_{ki}M, \forall i, j \in J, \forall k \in K \quad (10)$$

$$e_{ki} \leq y_{ki}M, \forall i, j \in J, \forall k \in K \quad (11)$$

$$M = \sum_{k, i, j} (T_{ii} + T_{kij} + 2nT_s), \forall i, j \in \bar{J}, \forall k \in K \quad (12)$$

目标函数(3)表示最小化作业完成时间; 式(4)表示 ASC 作业任务的先后关系, 每一台 ASC 从虚拟开始任务作业, 在虚拟结束任务终止; 式(5)表示每一个任务只能由一台 ASC 作业; 式(6)表示任务 j 在任务 i 之后由 k 作业; 式(7)表示每个任务的出度和入度; 式(8)表示任务开始作业的时间、运行时间以及作业完成时间之间的先后关系; 式(9)表示任务之间作业的先后时间关系, 并消除子回路; 式(10)、(11)为 ASC 作业的时间上限约束; 式(12)为 M 的取值.

为了规避干涉, 依据上述界定的 4 种干涉情况, 在模型 M 的基础上, 增加同步约束(13)~(22). 式(13)、(14)表示任务作业开始时间和完成时间; 式(15)~(18)界定出现干涉情况, 任务 i 或任务 j 其中一个完成时间较晚; 式(19)~(22)为干涉规避的同步约束, 表示当 ASC 规避时, 位于同一贝位的两个任务作业开始时间和完成时间之间的同步关系.

$$s_i = \sum_{k \in K} s_{ki}, \forall i \in J \quad (13)$$

$$e_i = \sum_{k \in K} e_{ki}, \forall i \in J \quad (14)$$

$$a_{ij} + a_{ji} = 1, \forall i, j \in J_s, L_i = L_j, i \neq j \quad (15)$$

$$b_{ij} + b_{ji} = 1, \forall i, j \in J_r, L_i = L_j, i \neq j \quad (16)$$

$$c_{ij} + c_{ji} = 1, \forall i \in J_s, \forall j \in J_r, L_i = L_j, i \neq j \quad (17)$$

$$d_{ij} + d_{ji} = 1, \forall i \in J_r, \forall j \in J_s, L_i = L_j, i \neq j \quad (18)$$

$$e_j \geq e_i + T_w + (a_{ij} - 1)M, \forall i, j \in J_s, L_i = L_j, i \neq j \quad (19)$$

$$s_j \geq s_i + T_w + (b_{ij} - 1)M, \forall i, j \in J_r, L_i = L_j, i \neq j \quad (20)$$

$$s_j \geq e_i + T_w + (c_{ij} - 1)M, \forall i \in J_s, \forall j \in J_r, L_i = L_j, i \neq j \quad (21)$$

$$e_j \geq s_i + 2T_w + (d_{ij} - 1)M, \\ \forall i \in J_r, \forall j \in J_s, L_i = L_j, i \neq j \quad (22)$$

3 算法设计与实现

3.1 算法设计

跨越式双 ASC 的调度问题属于 NP-hard 问题^[25],精确算法难以计算出最优解.遗传算法具有快速随机搜索机制,鲁棒性好,易于与其他算法(或策略)结合.因此,采用基于遗传算法求解任务分配与 ASC 的作业序列.首先,将装卸任务随机分配给 ASC;接着对 ASC 的作业序列做选择、交叉与变异操作;通过多次迭代搜索较为满意的解,直到满足停止条件,算法终止.由于 ASC 的作业过程受式(15)~(22)约束的制约,算法生成的作业序列存在不可行解.为解决这一问题,根据染色体上基因位对应的任务分配结果,按照任务优先级依次解码.解码策略为:依据贪婪规则将具有较短完成时间的任务优先插入 ASC 的作业序列中,并在干涉发生时做规避处理.将优先规则与遗传算法结合,执行交叉和变异操作的概率以随机实数作为参照,从而得到基于任务优先级的随机键遗传算法(PRK-GA),具体如下:

算法 1 基于任务优先级的随机键遗传算法(PRK-GA)

步骤 1 初始化种群 P .其中 P 包含 n 个染色体 $C_i, i \in \{1, 2, \dots, n\}$,每个染色体由两段基因构成,分别表示每台 ASC 分配的任务及其作业序列.

步骤 2 解码染色体 C_i .计算每个染色体对应的适应度(任务的完成时间).

步骤 3 选择适应度较高的染色体,复制到下一代中并组成新的种群 P^* .

步骤 4 以新种群 P^* 为父代,按照一定的交叉概率 p_c 、变异概率 p_m 执行交叉与变异操作,得到 P^* 的子代.

步骤 5 将得到的子代添加到种群中,更新种群 P .若满足停止条件,算法终止;否则转到步骤 2.

(1)染色体编码.任务依据初始顺序作优先级排序.采用十进制方式编码,若 n 个任务分配给 K 台 ASC 作业($K=2$),则染色体的编码长度为 $n+K$.其中实数 $1, 2, \dots, n$ 表示任务,带 # 的实数(1#, 2#)表示 ASC.因此,染色体可分为两段,每段染色体对应每台 ASC 所分配的作业任务集

合,集合中从左到右排列的顺序表示该 ASC 作业的任务序列.如图 3 所示,以两台 ASC 执行 9 个任务为例,染色体长度为 11.其中,ASC1 段表示 1# 的作业序列 $S_1 = \{7, 1, 5, 8\}$,ASC2 段表示 2# 的作业序列 $S_2 = \{6, 3, 9, 2, 4\}$.

染色体	1#	7	1	5	8	2#	6	3	9	2	4
ASC1段	1#	7	1	5	8						
ASC2段						2#	6	3	9	2	4

图 3 染色体编码示意图

Fig. 3 Diagram of chromosome coding

(2)适应度函数.目标函数为最小化最大完成时间 $f = \min(\max_i e_i)$,将种群中个体按照升序排序.在算法中通过缩放算法确定最终适应度,定义个体 m 的适应度为 $f(m)$.

(3)选择操作.采用轮盘赌方法,选择每次迭代后适应度最大的染色体复制到下一代.其中,个体 m 被选中的概率 $p(m) = f(m) / \sum_{m=1}^n f(m)$.

(4)交叉和变异.交叉和变异的目的是提高算法的搜索能力.其中,分别在两段染色体上选择两个基因位执行交叉操作,提高算法在 ASC 作业序列上的寻优能力,交叉示意图如图 4(a)所示.变异算子采取随机调换两个基因位的方法,在两段染色体上分别随机选择一个基因位,交换两个 ASC 段的任务,产生新的个体,变异示意图如图 4(b)所示.

父染色体1	1#	7	1	5	8	2#	6	3	2	4
父染色体2	1#	2	8	7	4	2#	5	6	1	3
子染色体1	1#	5	8	7	2	2#	3	6	1	4
子染色体2	1#	8	1	5	4	2#	7	3	2	6

(a) 交叉操作

父染色体	1#	7	1	5	8	2#	6	3	2	4
子染色体	1#	7	3	5	8	2#	6	1	2	4

(b) 变异操作

图 4 交叉和变异示意图

Fig. 4 Diagram of crossover and mutation

3.2 基于优先规则的解码算法

ASC 在作业过程中,包含 3 个参量:作业的任务 i_k 、ASC 的位置 b_k 和作业完成时间 t_k .3 个参量随着作业任务不同而发生变化,ASC 每次结束一个任务的作业,状态改变一次.因此,用包含

3 个参量的状态向量表示 ASC 的状态, 即 $\mathbf{v}_k = (i_k \ b_k \ t_k)$, $k \in K$. 将 ASC 作业不同任务的过程看作状态的转移, 故 ASC 作业的过程可表述为: ASC 从起始任务开始(初始状态), 依次作业所分配的任务; 每完成一个作业任务, 状态转移一次(对应状态向量更新一次), 直到任务被作业结束, 状态不再发生转移. 设 J_k 为 ASC k 完成作业任务集合, Ψ 为 ASC 已作业任务的开始时间和完成时间集合, T_k 为 ASC 任务更新前的作业时间.

为满足双 ASC 同步调度的要求, 当两台 ASC 在同一贝位上作业时, 规避干涉防止冲突发生. ASC k 作业的开始时间和完成时间分别为 s_{ki} 和 e_{ki} , 假定任务 i 和 j 分别由 ASC1 和 ASC2 作业. 规避干涉的规则如下:

(1) 任务 i 和 j 为存箱作业, 若 $|e_{1i} - e_{2j}| < T_s$, 则延迟其中一个任务的完成时间;

(2) 任务 i 和 j 为取箱作业, 若 $|s_{1i} - s_{2j}| < T_s$, 则延迟其中一个任务的开始时间;

(3) 任务 i 为存箱作业, 任务 j 为取箱作业, 若 $|e_{1i} - s_{2j}| < T_s$, 则延迟任务 i 的完成时间或延迟任务 j 的开始时间;

(4) 任务 i 为取箱作业, 任务 j 为存箱作业, 若 $|s_{1i} - e_{2j}| < T_s$, 则延迟任务 i 的开始时间或延迟任务 j 的完成时间.

ASC 已完成的作业任务存入集合 J_h , 对应的开始时间 s_{ki} 和完成时间 e_{kj} 存入集合 Ψ_k , 集合 $\Psi_k = \{(s_{kj}, e_{kj}), (s_{kl}, e_{kl}), \dots, (s_{km}, e_{km})\}$, $j, l, m \in J_h$, $k \in K$, 其中集合 Ψ_k 中的元素用任务编号索引. ASC 作业任务 j 对应的开始时间和完成时间为 T_j .

ASC 的状态转移是刻画作业过程中的状态变化. ASC 从一个状态转移到另一个状态, 其状态向量对应更新. 在 ASC 作业完成一个任务的同时, 对应新的状态向量 \mathbf{v}_k^* , 状态向量由 \mathbf{v}_k 转移到 \mathbf{v}_k^* . 式(23)为状态转移函数, 界定新的状态向量 \mathbf{v}_k^* 与任务 i 、执行该任务的 ASC k 以及更新前的状态向量 \mathbf{v}_k 之间的关系; 式(24)为位置转移函数, 界定新的位置 b_k^* 与任务 i 、ASC k 以及状态向量更新前的位置 b_k 之间的关系; 式(25)为完成时间转移函数, 界定新的时间 t_k^* 与任务 i 、ASC k 以及状态向量更新前的时间 t_k 之间的关系.

$$\mathbf{v}_k^* = f(k, \mathbf{v}_k, i) \quad (23)$$

$$b_k^* = f_b(k, b_k, i) \quad (24)$$

$$t_k^* = f_t(k, t_k, i) \quad (25)$$

为了说明函数内部执行规则与外部接口(输入、输出), 对状态转移函数 $\mathbf{v}_k^* = f(k, \mathbf{v}_k, i)$ 进行如下描述.

输入: 更新前的状态向量 \mathbf{v}_k

输出: 新的状态向量 \mathbf{v}_k^*

对应法则:

步骤 1 $i_k \leftarrow i, b_k \leftarrow D_i, t_k \leftarrow T_k + |D_i - O_i| \times T_b + 2T_s$.

步骤 2 写入 3 个元素更新后的状态向量到 \mathbf{v}_k^{**} , 即 $\mathbf{v}_k^{**} = (i_k \ b_k \ t_k)$.

步骤 3 更新 k 的状态向量 $\mathbf{v}_k \leftarrow \mathbf{v}_k^{**}$.

步骤 4 $\mathbf{v}_k^* \leftarrow \mathbf{v}_k$, 清除 \mathbf{v}_k^{**} .

解码算法根据任务优先级依次对染色体进行解码. 基于以上 4 条干涉规避规则与 3 个状态转移函数定义以及 3.1 节染色体的编码方式进行解码操作, 设 S_k 为 ASC 作业序列, n 为基于优先级排序的任务序列. 判断发生干涉的条件是: (1) 当前任务的开始时间(或完成时间)与已完成分配的任务开始时间和完成时间作比较(取时间差), 若存在时间差小于允许的时间长度 T_s , 则存在发生碰撞的可能, 但需要进一步判断这种可能是否位于同一贝位; (2) 若位于同一贝位, 则说明干涉发生, 令当前任务等待(延迟其作业的开始时间或完成时间), 实现规避. 对应于算法 2 的步骤 4 和步骤 5. 解码算法如下:

算法 2 基于任务优先级的解码算法

输入: n, K

输出: S_k

初始化: ASC 的作业序列 $S_k = \emptyset, \forall k \in K$; ASC 状态向量 $\mathbf{v}_k = (i_k \ b_k \ t_k) = (0 \ 0 \ 0), k = 1$; $\mathbf{v}_k = (i_k \ b_k \ t_k) = (0 \ 41 \ 0), k = 2$

步骤 1 当 $n = \emptyset$ 时, 算法终止; 否则, 转到步骤 2.

步骤 2 $i \leftarrow \text{POP}(n)$. 计算 $\mathbf{v}_k^* = f(k, \mathbf{v}_k, i)$, $\forall k \in K$, 取 $k = \arg \min_h t_h^*$. 把任务 i 添加到 ASC k 的作业序列中, 即 $S_k \leftarrow S_k \cup \{i\}$.

步骤 3 计算 $b_k^* = f_b(k, b_k, i), t_k^* = f_t(k, t_k, i)$. 计算 $\Psi_h, h = K \setminus \{k\}$.

步骤 4 干涉的存在条件判断. 若 $|t_k^* - \Psi_h(j)| < T_s, j \in J_h$, 取 $j = \arg(|t_k^* - \Psi_h(j)| < T_s)$; 若 $|t_k - \Psi_h(j)| < T_s, j \in J_h$, 取 $j = \arg(|t_k - \Psi_h(j)| < T_s)$, 转到步骤 5. 否则, 更新 $\mathbf{v}_k \leftarrow \mathbf{v}_k^*$, 转到步骤 1.

步骤 5 若 $O_{bl} = D_{bl}, l \in \{i, j\}, t_k^* \leftarrow t_k^* + T_w$, 更新 $v_k \leftarrow v_k^*$, 转到步骤 1.

4 计算实验

4.1 算例设计与分析

装卸任务在单箱区内进行,同一规格集装箱均匀分布,两端的交接区容量无限制.贝位数量 $B_n = 40$, 即贝位集合 $B = \{0, 1, \dots, B_n, 41\}$, 其中贝位 0 与 41 分别为两端交接区的位置. 1# 和 2# 可相互穿越,假定 ASC 每秒移动 4 个贝位,吊具提升或释放一个集装箱所花费的时间为 30 s,作业过程中速度恒定.设 ASC 移动一个贝位的时间 T_b 为单位时间. ASC 在列之间的移动时间和吊具在层之间的移动时间忽略不计.开始作业时, 1# 与 2# 的起始位置位于交接区,分别为 0 和 41 贝位.将待存集装箱从交接区运输到对应存箱位,将待取集装箱从箱区运输到两端交接区,最小化任务完成时间.

下面说明算例的生成规则.每个任务包括 3 个参数:存取作业 P_i 、起始位置 O_i 和目标位置 D_i .存取作业参数 P_i 随机生成, $P_i = 1$ 为取箱任务, $P_i = 0$ 为存箱任务.若 $P_i = 1$,起始位置 O_i 服从均匀分布 $U(1, 40)$,目标位置 D_i 可选 0 或 41;若 $P_i = 0$,起始位置 O_i 可选 0 或 41,目标位置 D_i 服从均匀分布 $U(1, 40)$.根据作业任务的数量,分别产生 5 组算例 I_{10} 、 I_{20} 、 I_{50} 、 I_{100} 、 I_{200} ,算例下标为任务的规模.其中,算例 I_{20} 如表 1 所示.模型 M 中其他参数如 L_i 、 T_{kij} 、 T_{ii} 可通过算例生成的参数求解.另外,设置等待时间 T_w 为提升或释放一个集装箱的服务时间,即 $T_w = T_s = 30 \text{ s} = 7.5 T_b$.

表 1 算例 I_{20}
Tab. 1 Example I_{20}

编号	P_i	O_i	D_i	编号	P_i	O_i	D_i
0	0	0	8	10	0	41	38
1	1	31	41	11	1	22	41
2	1	11	0	12	1	16	0
3	0	0	1	13	0	41	34
4	1	22	41	14	1	31	0
5	0	41	19	15	0	41	3
6	1	13	0	16	0	0	11
7	0	0	20	17	0	0	26
8	1	19	41	18	0	0	1
9	1	12	0	19	1	13	0

4.2 实验设置

使用 Gurobi 求解模型 M, PRK-GA 由 C++ 编译在 Visual Studio 2019 平台实现. PRK-GA 进化最大迭代次数为 300,迭代的终止条件为达到最大迭代次数.设计 4 组实验分别确定遗传算法参数设置的合理性、分析模型与算法的有效性、参数灵敏度以及测试算法的性能.

实验 1 确定 PRK-GA 算法参数设置.不同的参数设置将影响算法求解的精度和速度,针对这一问题,选取算例 I_{50} ,分别对种群规模 P 、交叉概率 p_c 和变异概率 p_m 在不同取值下进行实验,确定最终的参数设置.首先,确定种群规模.取 $p_c = 0.7, p_m = 0.1$,设置种群数量为 10~100,步长为 10;其次,确定 p_c 和 p_m 最佳组合,基于已确定的种群规模, p_c 和 p_m 分别依次取 0.1、0.3、0.5、0.7,得到 16 种组合方式.

实验 2 验证模型的正确性与算法的有效性. ASC1 和 ASC2 分别从起始位置开始作业,在同步约束下进行任务分配,并优化 ASC 作业序列,最小化最大作业完成时间.限于模型 M 求解的复杂性,用算例 I_{10} 作为测试集验证模型的正确性.为验证算法的性能,选取算例 I_{20} 进行分析. $I_{20-\alpha}$ 表示不同任务数量的算例,其中 α 取值为 1~20.

实验 3 分析同一贝位任务数量对结果的影响.同一贝位中任务数量越多,发生干涉的可能性越大.重构算例 I_{10} ,令 ω 为位于同一贝位任务的数量, ω 分别取 0~5 的正整数.分析不同 ω 下任务完成时间、空载时间和计算时间的变化趋势.

实验 4 测试并分析算法求解的质量.依次求解 5 组算例,计算得到任务分配与 ASC 作业序列,并展示部分算例的作业轨迹,分析是否成功地规避干涉、避免冲突的发生.

4.3 结果与分析

(1)实验 1 结果与分析.算法在不同种群规模下求解的结果如表 2 所示,当种群规模 $P = 80$ 时,完成时间最短,以下实验的种群规模设定为 80.交叉概率 p_c 和变异概率 p_m 的组合实验结果如表 3 所示,结果显示,当 $p_c = 0.7, p_m = 0.1$ 时,算法的寻优效果最好,求解速度较快.综上,算法的参数设置为 $P = 80, p_c = 0.7, p_m = 0.1$.

(2)实验 2 结果与分析.模型 M 与算法的计算结果对比如表 4 所示,当任务数不超过 10 时,

表 2 种群规模实验

Tab. 2 Experiment for population size

种群规模	完成时间/ T_b	种群规模	完成时间/ T_b
10	1 257	60	1 275
20	1 257	70	1 261
30	1 264	80	1 249
40	1 271	90	1 254
50	1 259	100	1 277

表 3 交叉概率和变异概率组合实验

Tab. 3 Combined experiment for probability between crossover and mutation

编号	p_c	p_m	完成时间/ T_b	计算时间/s
1	0.1	0.1	1 264	2.95
2	0.1	0.3	1 260	2.95
3	0.1	0.5	1 257	2.84
4	0.1	0.7	1 257	2.96
5	0.3	0.1	1 260	2.95
6	0.3	0.3	1 261	2.86
7	0.3	0.5	1 275	3.04
8	0.3	0.7	1 270	3.17
9	0.5	0.1	1 259	2.90
10	0.5	0.3	1 264	2.81
11	0.5	0.5	1 277	2.93
12	0.5	0.7	1 264	2.99
13	0.7	0.1	1 249	2.86
14	0.7	0.3	1 264	2.95
15	0.7	0.5	1 257	2.89
16	0.7	0.7	1 253	3.10

Gurobi 与 PRK-GA 求解结果基本一致. 当任务数大于 10 时, Gurobi 的求解时间较长, 个别解的质量稍差于 PRK-GA; 而 PRK-GA 能够在较短时间内求解出较为满意的解. 算例 I_{10} 求解结果对应的作业轨迹如图 5、6 所示. 在 119.5 时刻, 作业 7 号任务的 ASC1 和作业 5 号任务的 ASC2 (均为存箱作业) 在 19 贝位相遇, 由于任务的目标位置不同, 不发生干涉, ASC1 穿越通过正在执行释放集装箱操作的 ASC2, 如图 5 的 A 点, 实现干涉的成功规避. 以上结果, 验证了模型的正确性与 PRK-GA 的有效性.

(3) 实验 3 结果与分析. 同一贝位中任务数量越多, 发生干涉的可能性越大. 不同任务 i 下完成时间、空载时间和计算时间 (算法运行时间) 的变化如图 7 所示. 总体来说, 随着任务 i 的增加, 完成时间呈现上升趋势, 空载时间变化不大, 但计算时间变化明显. 由此可知, 位于同一贝位的任务数

表 4 Gurobi 与 PRK-GA 计算结果对比

Tab. 4 Solution comparison between Gurobi and PRK-GA

算例	Gurobi		PRK-GA		相对误差/%
	完成时间/ T_b	计算时间/s	完成时间/ T_b	计算时间/s	
I_{20-1}	23.00	0.01	23.00	0.12	0
I_{20-2}	35.00	0.01	35.00	0.12	0
I_{20-3}	49.00	0.01	52.00	0.15	6.12
I_{20-4}	66.00	0.07	68.00	0.12	3.03
I_{20-5}	88.00	0.08	88.00	0.27	0
I_{20-6}	106.00	0.52	125.00	0.24	17.92
I_{20-7}	122.00	1.06	125.00	0.21	2.46
I_{20-8}	140.00	5.45	143.00	0.20	2.14
I_{20-9}	161.00	86.38	162.00	0.24	0.62
I_{20-10}	175.00	621.10	178.00	0.14	1.71
I_{20-11}	196.00	3 600.00	180.00	0.17	-8.16
I_{20-12}	220.00	3 600.00	230.00	1.29	4.55
I_{20-13}	240.00	3 600.00	230.00	2.02	-4.17
I_{20-14}	251.00	3 600.00	283.00	2.19	12.75
I_{20-15}	278.00	3 600.00	302.00	1.27	8.63
I_{20-16}	301.00	3 600.00	374.00	1.11	24.25
I_{20-17}	314.00	3 600.00	374.00	1.86	19.11
I_{20-18}	329.00	3 600.00	380.00	1.44	15.50
I_{20-19}	348.00	3 600.00	393.00	1.29	12.93
I_{20-20}	370.00	3 600.00	421.00	1.34	13.78

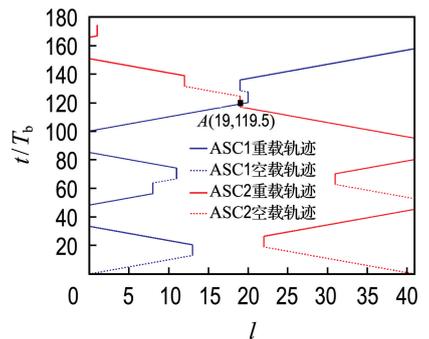


图 5 Gurobi 求解结果

Fig. 5 Solution by Gurobi

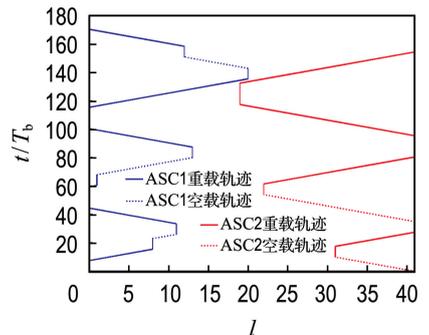


图 6 PRK-GA 求解结果

Fig. 6 Solution by PRK-GA

量越多,完成时间越长($i=3$ 除外),求解计算耗时明显增多.因此,在制订调度计划时,码头可通过调整位于同一贝位的任务数量,以减少任务的完成时间和计算耗时.

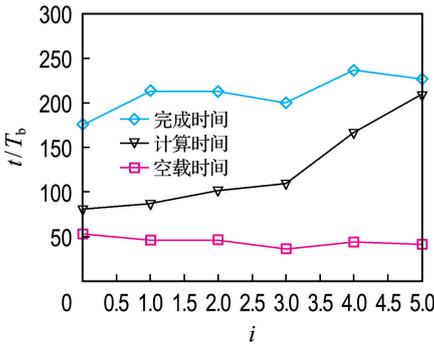


图 7 同一贝位任务数量对结果的影响

Fig. 7 The effect on solution by the number of job on the same bay

(4) 实验 4 结果与分析. PRK-GA 在满足同步约束的要求下,计算得出规避干涉后的结果,展示部分算例的作业轨迹.图 8 展示了算例 I_{20} 的作业轨迹,算法在 B 点和 C 点成功地规避了干涉.以 B 点为例说明规避的过程:两台 ASC 在 0 号贝位均待执行提取集装箱操作,由于开始时间差小于安全时间而产生干涉,为规避干涉,要求 ASC1 等待 ASC2 完成在该贝位的操作后再开始作业,即 ASC1 延迟 T_w 后再开始作业,B 点上下两侧箭头所夹的时间长度为 T_w .在 C 点,ASC1 执行存箱操作,ASC2 执行取箱操作,在 41 号贝位发生干涉,规避干涉的过程与 B 点类似,这里要求 ASC2 延迟 T_w 再执行取箱操作.

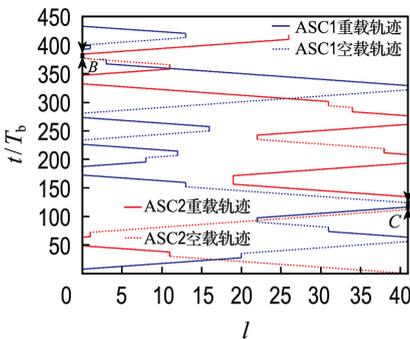


图 8 算例 I_{20} 的作业轨迹

Fig. 8 Operation track in example I_{20}

PRK-GA 在求解算例 I_{50} 时出现了 4 次干涉,结果显示干涉得到成功规避(见图 9).在干涉成

功规避的前提下完成了所有任务的分配与作业,保证了解的质量,同时验证了 PRK-GA 算法的有效性.

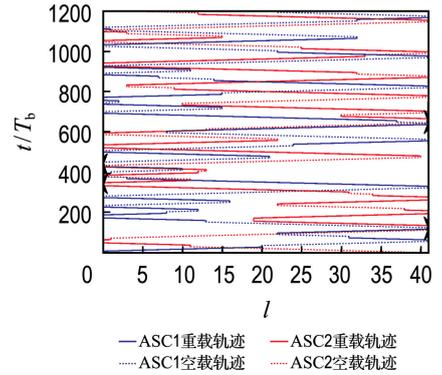


图 9 算例 I_{50} 的作业轨迹

Fig. 9 Operation track in example I_{50}

5 结 论

(1) 通过分析干涉发生的时间关系,界定了 4 种可能发生在同一贝位的干涉情况:①任务 i 和 j 都为存箱作业,两者的完成时间相同;②任务 i 和 j 都为取箱作业,两者的开始时间相同;③任务 i 为存箱作业,任务 j 为取箱作业,任务 i 的完成时间等于任务 j 的开始时间;④任务 i 为取箱作业,任务 j 为存箱作业,任务 i 的开始时间等于任务 j 的完成时间.处理以上情况的策略为:延迟其中一个任务作业的开始时间(或完成时间),使其等待另一个任务完成提取(或释放)集装箱的操作后再开始执行操作,避免两台 ASC 发生碰撞.模型中的同步约束是该策略的数学表达.

(2) 根据干涉规避策略,在干涉规避同步约束下,以最小化任务完成时间为目标,建立跨越式双 ASC 同步调度优化模型,确定每个任务的分配结果与 ASC 的作业序列.算例实验验证了模型的正确性和有效性.

(3) 为提升同步约束下的双 ASC 调度优化问题的求解效率,提出了基于任务优先级的随机键遗传算法.在染色体解码环节,用状态转移函数描述 ASC 的作业过程,采用基于优先级规则的解码策略,确定任务分配与 ASC 作业的序列.通过对求解结果的分析,验证了算法在规避干涉上的成效以及算法的求解性能,同时说明了算法的有效性.

事实上,箱区两端交接区容量有限,调度计划的制订应考虑交接位有限这一约束,进一步完善

堆场装卸设备调度理论. 因此, 下一步工作在干涉规避的基础上, 考虑交接区容量约束的穿越式双ASC调度问题.

参考文献:

- [1] DELL R F, ROYSET J O, ZYNGIRIDIS I. Optimizing container movements using one and two automated stacking cranes [J]. **Journal of Industrial and Management Optimization**, 2009, **5**(2): 285-302.
- [2] SAANEN Y A, VALKENGOED M V. Comparison of three automated stacking alternatives by means of simulation [J]. **Proceedings - Winter Simulation Conference**, 2005, **2005**: 1567-1576.
- [3] VIS I F A, CARLO H J. Sequencing two cooperating automated stacking cranes in a container terminal [J]. **Transportation Science**, 2010, **44**(2): 169-182.
- [4] 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.
- [5] STAHLBOCK R, VOß S. Operations research at container terminals; a literature update [J]. **OR Spectrum**, 2008, **30**(1): 1-52.
- [6] 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.
- [7] CARLO H J, VIS I F A, ROODBERGEN K J. Transport operations in container terminals; Literature overview, trends, research directions and classification scheme [J]. **European Journal of Operational Research**, 2014, **236**(1): 1-13.
- [8] BOYSEN N, BRISKORN D, MEISEL F. A generalized classification scheme for crane scheduling with interference [J]. **European Journal of Operational Research**, 2017, **258**(1): 343-357.
- [9] 胡祥培, 孙玉姣, 曾庆成. 集装箱码头同贝同步装卸作业的序列优化模型 [J]. **系统工程理论与实践**, 2016, **36**(3): 623-634.
HU Xiangpei, SUN Yujiao, ZENG Qingcheng. A sequence optimization model of dual-cycling operations in container terminals [J]. **Systems Engineering - Theory & Practice**, 2016, **36**(3): 623-634. (in Chinese)
- [10] VIS I F A, ROODBERGEN K J. Scheduling of container storage and retrieval [J]. **Operations Research**, 2009, **57**(2): 456-467.
- [11] GHAREHGOZLI A H, LAPORTE G, YU Yugang, *et al.* Scheduling twin yard cranes in a container block [J]. **Transportation Science**, 2015, **49**(3): 686-705.
- [12] HE Junliang, HUANG Youfang, YAN Wei. 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.
- [13] LI Wenkai, WU Yong, PETERING M E H, *et al.* Discrete time model and algorithms for container yard crane scheduling [J]. **European Journal of Operational Research**, 2009, **198**(1): 165-172.
- [14] 魏晨, 胡志华, 高超锋, 等. 自动化集装箱码头堆场内双起重机调度模型与算法 [J]. **大连海事大学学报**, 2015, **41**(4): 75-80, 89.
WEI Chen, HU Zhihua, GAO Chaofeng, *et al.* Scheduling model and algorithm of twin synchronized stacking cranes in stack yard of automated container terminal [J]. **Journal of Dalian Maritime University**, 2015, **41**(4): 75-80, 89. (in Chinese)
- [15] 周静娴, 胡志华. 自动化集装箱码头穿越式双自动堆码起重机调度优化 [J]. **计算机应用**, 2015, **35**(9): 2673-2677.
ZHOU Jingxian, HU Zhihua. Scheduling optimization for cross-over twin automated stacking cranes in automated container terminal [J]. **Journal of Computer Applications**, 2015, **35**(9): 2673-2677. (in Chinese)
- [16] HU Zhihua, SHEU J B, LUO J X. Sequencing twin automated stacking cranes in a block at automated container terminal [J]. **Transportation Research Part C: Emerging Technologies**, 2016, **69**: 208-227.
- [17] 范厚明, 马梦知, 姚茜, 等. 集装箱堆场箱位分配及多场桥调度协同优化问题 [J]. **上海交通大学学报**, 2017, **51**(11): 1367-1373.
FAN Houming, MA Mengzhi, YAO Xi, *et al.* Integrated optimization of storage space allocation and multiple yard cranes scheduling in a container terminal yard [J]. **Journal of Shanghai Jiao Tong University**, 2017, **51**(11): 1367-1373. (in Chinese)
- [18] ZHOU Weihua, WU Xiaobo. An efficient optimal solution of a two-crane scheduling problem [J]. **Asia-Pacific Journal of Operational Research**, 2009,

- 26(1): 31-58.
- [19] BRISKORN D, EMDE S, BOYSEN N. Cooperative twin-crane scheduling [J]. **Discrete Applied Mathematics**, 2016, **211**: 40-57.
- [20] HAN Xiaolong, WANG Qianqian, HUANG Jiwei. Scheduling cooperative twin automated stacking cranes in automated container terminals [J]. **Computers and Industrial Engineering**, 2019, **128**: 553-558.
- [21] BRISKORN D, ANGELOUDIS P. Scheduling cooperating stacking cranes with predetermined container sequences [J]. **Discrete Applied Mathematics**, 2016, **201**: 70-85.
- [22] NOSSACK J, BRISKORN D, PESCH E. Container dispatching and conflict-free yard crane routing in an automated container terminal [J]. **Transportation Science**, 2018, **52**(5): 1059-1076.
- [23] DORNDORF U, SCHNEIDER F. Scheduling automated triple cross-over stacking cranes in a container yard [J]. **OR Spectrum**, 2010, **32**(3): 617-632.
- [24] 冯媛君, 曾庆成, 李厚源, 等. 集装箱码头堆场翻箱与外集卡提箱顺序同步优化方法 [J]. **运筹与管理**, 2017, **26**(5): 110-118.
FENG Yuanjun, ZENG Qingcheng, LI Houyuan, *et al.* Method for integrated optimizing of rehandling and retrieving sequence for inbound containers in container terminals [J]. **Operations Research and Management Science**, 2017, **26**(5): 110-118. (in Chinese)
- [25] NG W C. Crane scheduling in container yards with inter-crane interference [J]. **European Journal of Operational Research**, 2005, **164**(1): 64-78.

Synchronous scheduling optimization for dual automated stacking crane with interference at automated container terminals

WANG Yaozong, HU Zhihua*

(Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China)

Abstract: Aiming at the interference problem in the operation of dual automated stacking crane, the existing forms of interference are analyzed and defined. A mixed integer programming model for synchronous scheduling of dual automated stacking crane is established to determine the distribution of handling jobs and the sequence of each crane under the minimization of the makespan. The handling process of each crane is depicted by state transition function. A priority-key genetic algorithm based on task priority is proposed to solve the proposed optimization problem. Several numerical examples with various scales are designed according to the feature parameters at real-world automated container terminals for numerical experiment. The experimental results indicate that the proposed model satisfies the constraints of synchronous scheduling. All the solutions are conflict-free scheduling, which verifies the validity of the model. The results show that the solution of solving by proposed algorithm can avoid interference and achieve synchronous scheduling. The reasonability and validity of the algorithm are verified. In addition, this algorithm can effectively improve the solving rate of optimization of scheduling for dual automated stacking crane with synchronous constraints.

Key words: automated container terminal; scheduling of dual cranes; interference; priority; genetic algorithm