

基于遗传算法的自动化集装箱码头双轨道吊协同调度优化研究

黄继伟 韩晓龙

(上海海事大学物流研究中心 上海 201306)

摘要 为解决自动化集装箱码头中堆场并行式双自动化轨道吊存取箱作业冲突问题,考虑双轨道吊作业干扰约束,以任务完成总时间最小化为目标,建立双轨道吊协同调度的混合整数规划模型。设计基于作业顺序编码的遗传算法,并通过多组实验对 CPLEX 和遗传算法进行对比分析。实验结果表明,两种方法均可有效避免干扰发生并且都能找到最优接力区位置,充分论证了该双轨道吊协同调度优化模型及算法的有效性。同时算法性能评估表明,遗传算法更适合求解大规模设备调度问题。通过对遗传算法各参数进行优化,显著提高了求解效率。

关键词 自动化集装箱码头 自动化轨道吊 协同调度 干扰 遗传算法 混合整数规划

中图分类号 TP391.9 U691.3 **文献标识码** A **DOI:** 10.3969/j.issn.1000-386x.2018.09.016

COLLABORATIVE SCHEDULING OPTIMIZATION OF TWIN AUTOMATED STACKING CRANES IN AUTOMATIC CONTAINER TERMINALS BASED ON GENETIC ALGORITHM

Huang Jiwei Han Xiaolong

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

Abstract For the collaborative scheduling problem of the twin automatic stacking cranes (ASCs) with both of operations (storage and retrieval) on a single container block of the automated container terminals, a mixed-integer programming model was proposed to minimize the makespan of all the tasks. A genetic algorithm based on job sequence coding was designed and multiple sets of practical examples were solved by using CPLEX and genetic algorithm, considering the interferences between the twin ASCs. It is proved that both two methods can avoid the interferences efficiently between twin ASCs and find their optimal locations of the handshake area, which fully demonstrated the effectiveness of the proposed optimization model and algorithms. And the algorithm performance evaluation shows that genetic algorithm is more suitable for solving large-scale jobs and the optimization of parameters in genetic algorithm improved the solution efficiency significantly.

Keywords Automated container terminals Automated stacking cranes Collaborative scheduling Interferences Genetic algorithm Mixed integer programming

0 引言

随着经济全球化进程的加快和世界经济贸易高速发展,集装箱港口得到迅猛发展,各集装箱港口吞吐量持续快速增长。为满足集装箱船舶大型化和航运联盟化的需求,安全、高效、环保、节能的自动化集装箱码头 ACTs(Automated Container Terminals)备受青睐。建

设自动化集装箱码头成为全球各大港口的内在需求和发展趋势。

码头堆场作业是集装箱港口运作的重要部分。自动化集装箱码头的堆场各箱区一般配置两台相同的两端式自动化轨道吊 ASCs(Automatic Stacking Cranes)。两台轨道吊共用相同的轨道且不可互相穿越通过。轨道吊是集装箱港口作业的瓶颈,双 ASC 的协同调度直接影响到港口整体运行效率。在自动化集装箱码头文

献研究方面,文献[1]对堆场作业相关研究如装卸运输设备等进行了深入的概述,重点介绍了当前的行业趋势和发展状况,提出了堆场作业分类方法,并对2004年—2012年间发表的科研期刊论文进行了分类。文中还探讨了现有的堆场作业模式,最后根据集装箱码头行业的发展趋势指出新的研究点。码头堆场作业是集装箱港口运作的重要部分。在自动化集装箱码头中,自动化轨道吊ASCs的采用大幅提升了港口整体运行效率,轨道吊的调度策略直接影响到码头生产运营。文献[2]对集装箱码头的主要物流流程和运营情况的现有研究情况进行了详尽描述和分类,并对相关优化方法进行了全面调研。Vis等^[3]将自动化集装箱码头堆场内集装箱的存取问题视为农村邮路问题的一个特例,并将该问题重新定义为一个不对称的Steiner旅行商问题,以重构方式在部分问题二分网络中的最优分配以及这些部分之间连接的动态规划组合来有效解决ASC调度问题以实现优化。文献[4]全面研究了自动化集装箱码头堆场不同领域的路线规划中的各种问题。提出了基于集装箱装卸问题的数学分析模型,并考虑箱区长度、作业流畅度及安全距离等因素,开发了一种基于遗传算法的最优路径规划方法。最后通过实验算例验证了该算法的有效性,为提高集装箱装卸效率探索出了一条新途径。文献[5]针对自动化码头并行式双ASC调度问题首次对双ASC优先级进行对比分析,并量化了所选优先级的影响,提出了14条双ASC优先规则,并通过仿真对完工时间进行了评估。最后将优先级规则与使用分支定界法求得的最优优先级进行比较,得到最优的单调度规则和调度规则组合。文献[6]就集装箱码头并行式双ASC在海侧完成进口集装箱存箱作业的调度问题进行了探索,考虑双ASC之间的作业干扰和ASC优先权问题,以总完工时间最小化为目标,提出了基础问题的复杂性证明,并引入和测试了高效的启发式算法。文献[7]探讨了由双ASC合作完成装卸任务对任务执行效果的影响,开发启发式算法求解并通过算例对算法性能进行评估。文献[8]也针对作业任务已分配完成的情况探讨了两端式和穿越式两种轨道吊配置的优先作业顺序问题,建立相应的图形模型并开发强多项式算法进行求解。

针对堆场存箱或取箱的单一类型作业问题,文献[9]探讨了单箱区内海陆侧集装箱存取作业的双ASC调度问题,以双ASC完工时间最小化为目标,建立整数规划模型并提出了两种算法对该问题进行求解。文献[10]就洋山四期自动化集装箱码头的双ASC协同调度问题进行了研究。通过分析任意两个任务之间的最小时间间隔来量化ASC之间的干扰,建立了三个数

学模型,以最小化总完工时间对集装箱作业任务进行排程。设计精确算法和遗传算法来解决该问题,同时就所提出的模型和算法对实际自动化集装箱码头的管理理念和技术方面的应用进行了探讨。文献[11]针对堆场同一箱区内存箱或取箱的单一作业,考虑双起重机时空同步约束条件,以最小化作业总完成时间为目标建立双起重机调度混合整数规划模型,并设计遗传算法对大规模任务数量问题进行求解。算例分析结果表明,在大规模问题上遗传算法在解的质量上逐渐优于CPLEX算法且运算时间远小于CPLEX,证明了该双起重机调度模型与算法的有效性及合理性。

针对接力区或缓冲区的设计问题,文献[12]探索了自动化集装箱码头海陆侧接力区的设计对双自动化轨道吊作业性能的影响。以任务总完工时间或因作业干扰造成的ASC等待时间最小化为目标,通过变换堆场设计条件如ASC作业方式、接力区中集装箱对方位置以及接力区的有无、大小、数量等,并针对不同条件开发多种启发式方法进行求解。文献[13]通过对成对自动堆垛起重机在堆场实时调度问题的研究。建立了以最小化外集卡和船舶延时为目标的不对称的多旅行商模型,引入了接力、缓冲、干扰等关键约束,并利用并行实时调度策略对多组情景进行案例研究分析。结果表明设置主缓冲区以及接力区可以减少作业延时且海侧作业量较多时,增加缓冲区的容量可以更有效地增加作业效率。

现有研究多只针对堆场存箱或取箱操作的单一作业类型进行探讨,而对于单箱区内考虑存箱和取箱作业同时进行以及接力区的设计的研究相对较少。本文同时考虑进、出口箱的存取操作,以任务完成总时间最小化为目标,建立双轨道吊协同调度混合整数规划模型,有效避免ASC之间作业干扰,并用CPLEX和遗传算法对不同规模算例进行求解。

1 问题描述

目前,国际上自动化集装箱港口堆场多采用垂岸式布局,双ASC协同作业完成对进出于堆场的各种集疏港和装卸船的集装箱的装卸操作。堆场箱区布局如图1所示。当码头装卸任务下达后,分配到各箱区的集装箱需要在最短的时间内处理完毕,这对各箱区内设备的集装箱处理能力提出了更高的要求。两台相同的轨道吊在完成各自的装卸任务时往往发生作业干扰。若在箱区中设置接力区,使双ASC通过接力完成集装箱存取操作,即可大幅减少双ASC的作业干扰频率。因此,确定合适的接力区位置、有效避免双ASC

作业干扰成为优化堆场作业、提高堆场作业效率的关键。

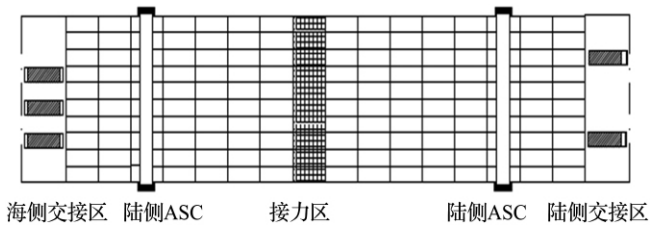


图1 自动化集装箱港口箱区布局示意图

为有效解决双 ASC 存取箱操作同时进行的协同调度干扰问题,采用以下任务拆分规则:箱区中设接力区(Handshake area),接力区大小为1贝。凡需要跨越接力区才能完成的任务均以接力区为界限进行拆分,即每个符合要求的初始任务均拆分为两个子任务,并分别由海陆两侧 SAC 协作完成存取操作。同一箱区内海、陆侧两端各设有一个集装箱交接区,又称为 I/O 点(I/O point),集港或疏港的外集卡与陆侧 ASC 在陆侧 I/O 点进行交接,对应的载有集装箱的自动导引车 AGVs(Automated Guided Vehicles)与海侧 ASC 在海侧 I/O 点进行交接。各 ASC 只能在各自半箱区内作业,跨越半箱区的作业任务均需双 ASC 利用接力区合作完成,双 ASC 之间的作业干扰只发生在接力区。假设箱区长度为 L 贝位,贝位数为 0 的箱位为海侧 ASC 与 AGV 的交接点,贝位数为 $L+1$ 的箱位为陆侧 ASC 与外集卡的交接点。海陆两侧 ASC 作业范围分别跨越半箱区,双 ASC 分别负责本侧集装箱的装卸作业。堆场作业任务主要分为存箱操作 S 型(Storage requests)和取箱操作 R 型(Retrieval requests)两种。将海侧 ASC 与 AGV 交接点和陆侧 ASC 与外集卡交接点均称为交接点,将存箱操作中集装箱的存放位置及取箱操作中集装箱的提取位置均称为目标点。故堆场 ASC 存箱和提箱作业过程均可简化为 4 个步骤,如表 1 所示。

表1 存箱和提箱操作步骤

步骤	存箱操作	取箱操作
1	从当前点移动至交接点	从当前点移动至目标点
2	在交接点提取集装箱	在目标点提取集装箱
3	将集装箱移动至目标点	将集装箱移动至交接点
4	在目标点存放集装箱	在交接点放下集装箱

引入时间-贝位坐标系,其中横轴为各任务的执行时刻,纵坐标为对应集装箱的位置,各操作步骤与图中各曲线段一一对应,如 ASC 无需移动即可完成存取操作则无对应线段。通过找到坐标中对应线段的分段函数图像交点可简明直观地描述出双 ASC 之间作业干

扰是否发生或发生在何处。双 ASC 作业干扰问题转化为所有子任务的时间-贝位曲线不相交问题。

符合拆分规则的初始任务经拆分后,子任务之间的干扰只发生在接力区位置,这样可以有效避免双 ASC 在执行任务时的相互干扰。任意两个紧前紧后关系的子任务 i, j 在接力区位置的干扰状态均满足图 2 中 4 种模式的一种。任务 i, j 分别由不同的 ASC 执行,各子任务一经确定,除 ASC 完成上一任务后移动至下一任务的衔接作业状态不确定外,其余三个连续步骤均随之确定。线段斜率为非负时表示该步骤中 ASC 处于运动状态,且移动速率为恒值 v ; 线段斜率为 0 时表示 ASC 正在执行存取操作,且各操作需固定作业时间 τ 。

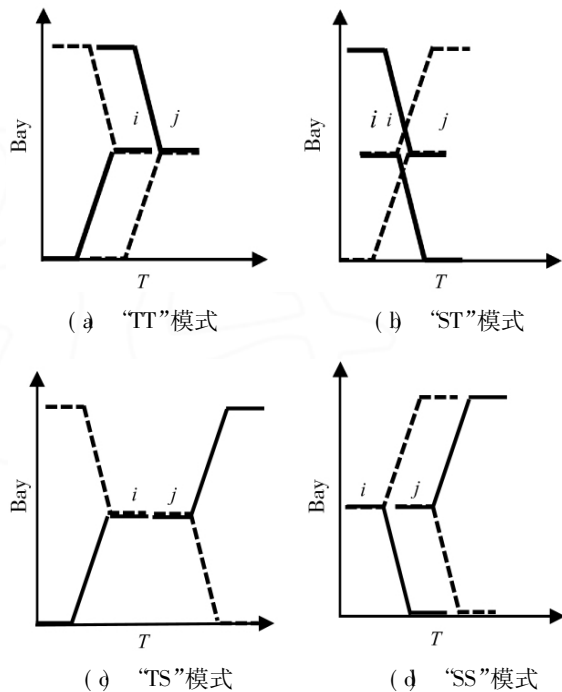


图2 有接力区设计下任务的4种干扰模式

每种干扰模式下子任务关系均可能存在两种情形,分别如图2中虚、实线图形所示。子任务的开始和结束状态分别用“S”和“T”表示,根据 ASC 在接力区位置执行前后任务的操作步骤。将(a)两种情形定义为“TT”干扰模式,表示双 ASC 发生干扰时其在接力区位置分别执行的任务 i, j 均处于结束状态;同理,将(b)两种情形定义为“ST”干扰模式,表示双 ASC 发生干扰时其在接力区位置分别执行的任务 i 处于开始状态,任务 j 处于结束状态;将(c)两种情形定义为“TS”干扰模式,表示双 ASC 发生干扰时其在接力区位置分别执行的任务 i 处于结束状态,任务 j 处于开始状态;将(d)两种情形定义为“SS”干扰模式,表示双 ASC 发生干扰时其在接力区位置分别执行的任务 i, j 均处于开始状态。

2 模型建立

为简化流程同时不失一般性,根据问题作以下假设:(1) ASC 在箱区两端装卸作业时不需等待;(2) 接力区存储空间足够大;(3) 双 ASC 匀速移动且速率相同;(4) 双 ASC 取箱和放箱的固定作业时间相同且为固定值;(5) 忽略 ASC 加速与减速及小车移动的影响。ASC 装卸作业满足以下要求:(1) 双 ASC 之间需保持一定的安全距离,通常为一个贝位;(2) 双 ASC 不能同时对同一集装箱执行装卸任务,且拆分后的子任务有明显的先后顺序,即一台 ASC 将集装箱移动至接力点并安全退出后,另一 ASC 方能对该集装箱执行下一阶段的操作。

相关符号说明:

J_k : 第 k 台 ASC 的任务集, $k=1, 2$;

$A = \{(i, j) \mid i \in J_k, j \in J_l, k \neq l\}$, 同一任务拆分后的子任务对 $\{i, j\}$ 的集合;

J : 所有 ASC 的任务集, $J = J_k \cup J_l$;

$B = \{0, \dots, L+1\}$: 贝位集, 其中 L 为箱区长度, 0 表示海侧 I/O 点的贝位, $L+1$ 表示陆侧 I/O 点的贝位;

B_s^i : 任务 i 的开始贝位, $i \in J$;

B_t^i : 任务 i 的结束贝位, $i \in J$;

exB : 双 ASC 作业接力区的贝位;

v : ASC 水平移动速率;

τ : ASC 提箱或存箱作业的固定作业时间;

T_{ij} : ASC 从任务 i 的结束位置到任务 j 的开始位置的移动时间, 特别地, 当 $i=j$ 时, T_{ij} 表示 ASC 从任务 i 的开始位置到其结束位置的移动时间;

T : 任务总完工时间。

决策变量定义如下:

s_i : 任务 i 的开始时刻, $i \in J$;

t_i : 任务 i 的结束时刻, $i \in J$;

$u_i \in \{0, 1\}$, 当 ASC 从任务 i 开始作业时 $u_i = 1$;

$v_i \in \{0, 1\}$, 当 ASC 在任务 i 结束作业时 $v_i = 1$;

$x_{ij} \in \{0, 1\}$, 任务 i, j 为紧前紧后关系且被同一 ASC 执行时 $x_{ij} = 1$;

$y_{ij} \in \{0, 1\}$, $i \in J_k, j \in J_l, k \neq l$, 任务 i 早于任务 j 被执行且 i 和 j 被不同 ASC 执行时 $y_{ij} = 1$;

σ : 两 ASC 作业的安全距离;

$T_i [B_s^i, B_t^i]$: 任务 i , 其中, B_s^i 和 B_t^i 分别表示任务 i 的起始贝位和终止贝位, 如 $T_3 [5, 40]$ 表示任务 3 要求将贝位为 5 的集装箱移动至贝位为 40 的 I/O 点, 该任

务为取箱作业(R型), 该集装箱可能为进口箱。

M 为一足够大的正数。

根据上述信息建立混合整数规划模型:

$$\text{Min } T = \max\{t_i\} \quad (1)$$

$$t_i \geq s_i + 2\tau + T_{ii} \quad \forall i \in J \quad (2)$$

$$s_j \geq t_i + T_{ij} + M(1 - x_{ij}) \quad \forall i, j \in J_k, k=1, 2 \quad (3)$$

$$\sum_i x_{ij} + u_j = 1 \quad \forall j \in J_k, k=1, 2 \quad (4)$$

$$\sum_j x_{ij} + v_i = 1 \quad \forall i \in J_k, k=1, 2 \quad (5)$$

$$T_{ij} = \frac{|B_s^j - B_t^i|}{v} \quad (6)$$

$$t_j - \tau \geq t_i + \sigma - M(1 - y_{ij})$$

$$\forall i \in J_k, \forall j \in J_l, k \neq l \text{ 且 } B_t^i = B_s^j \quad (7)$$

$$t_j - \tau \geq (s_i + \tau) + \sigma - M(1 - y_{ij})$$

$$\forall i \in J_k, \forall j \in J_l, k \neq l \text{ 且 } B_s^i = B_t^j \quad (8)$$

$$s_j \geq t_i + \sigma - M(1 - y_{ij})$$

$$\forall i \in J_k, \forall j \in J_l, k \neq l \text{ 且 } B_t^i = B_s^j \quad (9)$$

$$s_j \geq (s_i + \tau) + \sigma - M(1 - y_{ij})$$

$$\forall i \in J_k, \forall j \in J_l, k \neq l \text{ 且 } B_s^i = B_s^j \quad (10)$$

$$s_i \geq T_{0i} - M(1 - u_i) \quad \forall i \in J_k, k=1, 2 \quad (11)$$

$$s_j \geq t_i + \sigma \quad \forall \{i, j\} \in A \quad (12)$$

式(1)表示模型目标为最小化作业总完工时间。

约束条件式(2)表示同一任务的起始时间关系; 约束条件式(3)表示任意两个任务之间的起止时间关系; 约束条件式(4)、式(5)保证了同一 ASC 每次只能执行一个任务; 约束条件式(6)表示任务执行时间与执行该任务的 ASC 的位置关系; 约束条件式(7) - 式(10)分别表示在“TT”、“ST”、“TS”和“SS”4 种干扰模式下双 ASC 执行任意一组紧前紧后任务对 $\{i, j\}$ 时均须保持一定的安全距离; 约束条件式(11)是对作业初始化进行定义; 约束条件式(12)表示对于由同一任务拆分来的紧前紧后子任务对 $\{i, j\}$ 之间的作业先后关系。

对于大规模问题, 很难直接求解。下面给出遗传算法。

3 遗传算法

3.1 参数设置

某自动化集装箱港口垂岸式堆场采用两端式双 ASC 布置, 两台同型号的轨道吊 C_1 (即海侧 ASC) 和 C_2 (即陆侧 ASC) 协同完成作业任务。箱区长度为 39 贝位, 接力区初始位置为第 20 贝位, 第 0 贝位表示海侧 ASC 与 AGV 的交接点, 第 40 贝位为表示陆侧 ASC 与外集卡的交接点; 双 ASC 匀速移动且速度相同, 单位

时间(设为 1 s)内移动距离为 1 贝;ASC 取箱和放箱的固定作业时间为 3 s;两台轨道吊作业干扰只发生在接力区,双 ASC 间须保持安全距离 1 贝,显然在数值上有 $\delta = 1/v$ 。

3.2 算法设计

CPLEX 是求解线性规划和混合整数线性规划等问题的非常高效实用的工具,现利用 Python 调用 CPLEX 对该算例进行求解。对于需要拆分的初始任务 $T_i [B_s^i, B_f^i]$ 序列进行编码,序列 (a, b) 、 (a, c) 分别表示初始任务 a 经拆分后得到的子任务 b 和 c 的序列,如初始任务 $T_0 [17, 40]$ 经拆分后得到两个子任务 $(17, 20)$ 和 $(20, 40)$ 的序列编码分别为 $(0, 0)$ 和 $(0, 1)$ 。

遗传算法具有良好的全局搜索能力,它通过模仿自然界的选择与遗传的机理,利用迭代寻求全局最优,并利用其内在并行性,方便地进行分布式计算以加快求解速度,对作业调度类问题具有良好的适应性^[14]。

3.2.1 选择

适应度函数由最小化 ASC 作业完工时间的目标函数转化而来,并采用比较经典的锦标赛选择方法选择个体。

3.2.2 编码与解码

在对任务序列进行编号时,初始任务规模为 n ,则任务 i 与 $n+i$ 表示由同一初始任务 i 拆分而来的子任务,且 i 为紧前任务。例如当 $n=4$ (初始序列为 0, 1, 2, 3) 时,若任务 0, 2, 3 需要被拆分,则子任务序列为 $(0, 1, 2, 3, 4, 6, 7)$,如表 2 所示。当解码染色体时,由同一初始任务拆分得到的先出现的子任务实际被优先执行或在接力区享有优先权。

表 2 任务编号示例

任务序列	0	1	2	3	0	4	1	2	6	3	7
起点	17	27	6	0	17	20	27	6	20	0	20
终点	40	40	40	28	20	40	40	20	40	20	28

现有 10 个初始任务,根据任务拆分规则得到 18 个子任务,由两台 ASC 协作完成。染色体由两条组成,分别为任务段和 ASC 段,任务段中的基因分别对应各子任务,ASC 编号由各子任务被执行的情况而定, C_1 需完成 $[0, 2, 3, 6, 9, 11, 14, 17, 18]$, C_2 需完成 $[1, 4, 5, 7, 8, 10, 12, 13, 16]$ 。染色体编码解码情况如图 3 所示。

任务编号	0	1	2	...	14	16	17	18
ASC 编号	1	2	1	...	1	2	1	1

图 3 染色体编码解码示意图

3.2.3 交叉

交叉运算是产生新个体的主要方法,它决定了遗传算法的全局搜索能力,在遗传算法中起关键作用。本文采用 PMX 交叉方法,首先随机选择一对父代染色体中几个基因的起止位置,然后交换这两组基因的位置,最后做冲突检测,根据交换的两组基因建立一个映射关系。所有冲突的基因经过映射,形成新一对没有冲突的子代基因。PMX 交叉如图 4 所示。

父代 1	1	2	3	4	5	6	7	8	9
父代 2	5	8	6	9	2	1	7	4	3
↓									
子代 1	3	5	6	9	2	1	7	8	4
子代 2	2	8	3	4	5	6	7	9	1

图 4 染色体 PMX 交叉示意图

3.2.4 变异

染色体包含任务段和 ASC 段,选择两组等位基因进行变换。变异运算是产生新个体的辅助方法,它决定了遗传算法的局部搜索能力。变异运算示意图如图 5 所示。

父代任务段	3	1	2	5	7	8	6	4	9
父代 ASC 段	2	1	1	2	2	1	2	1	1
↓									
子代任务段	3	1	8	5	7	2	6	4	9
子代 ASC 段	2	1	1	2	2	1	2	1	1

图 5 染色体变异示意图

3.3 结果分析

3.3.1 算法性能分析

设置 6 组不同规模的算例对两种算法的性能进行评估,假设接力区处于箱区中间位置(第 20 贝位)。遗传算法求解时参数设置为种群规模 $N=200$,交叉概率 $cpxb=0.5$,变异概率 $mutpb=0.1$,迭代次数 $ngen=300$,使用相同的参数运算 10 次取平均值。两种方法求解结果如表 3 所示。

表 3 CPLEX 与遗传算法求解不同规模算例性能结果对比

序号	规模	CPLEX		GA				
		最优值/s	运行时间/s	最小值/s	最大值/s	平均值/s	平均用时/s	误差/%
1	4	146	0.02	146	146	146	42.1	0.00
2	10	234	637.53	236	245	239.5	99.6	-2.35
3	20	438	3 844	438	444	438.6	189.4	-0.14
4	50	1 076*	>9 999*	1 044	1 085	1 053.1	401.7	2.13
5	100	2 913*	>9 999*	2 275	2 286	2 275.3	684.5	21.9
6	200	6 722*	>9 999*	4 745	4 771	4 756.9	1 465.8	29.2

注:任务规模为 50 时 CPLEX 运行内存溢出;误差为 GA 所得最优解平均值相对 CPLEX 最优值的结果

求解过程中,当算例规模较大时,CPLEX 运行内存溢出,但仍可以找到可行解。由表3易知,两种算法性能对比发现,即使 CPLEX 的内存管理非常有效,但是求解较大规模的线性规划问题,内存不足仍然是需要考虑的问题之一,而遗传算法求解大规模算例具有独特优势。

3.3.2 双 ASC 作业干扰优化分析

当自动化集装箱码头的装卸任务下达后,分配到各箱区的集装箱需要在最短的时间内处理完毕,这对各箱区内设备的集装箱处理能力提出了更高的要求。两台相同的轨道吊在完成各自的装卸任务时往往发生作业干扰。若在箱区中设置接力区,使双 ASC 通过接力完成集装箱存取操作,即可大幅降低双 ASC 的作业干扰频率。因此,确定合适的接力区位置、有效避免双 ASC 作业干扰成为优化堆场作业、提高堆场作业效率的关键。针对第2组实例运用 CPLEX 和遗传算法对该问题进行求解,并评估两种方法对堆场设备协同调度的优化效果。当变换接力区位置时,求得的不同接力区位置对应的最小总完工时间结果如图6所示。

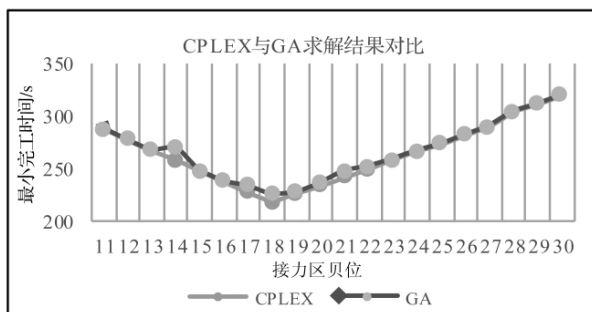


图6 CPLEX 与 GA 求解结果对比

由图6可知,两种方法所求接力区最优位置均为第18贝位。CPLEX 求得的最小总完工时间为218 s,双 ASC 在作业过程中未发生干扰,求解历时637.53 s;采用遗传算法,每组重复10次运算后取最优值,求得的最小总完工时间为225 s,双 ASC 在第151 s发生一次干扰,平均求解历时86.033 s。两种算法求得的双 ASC 作业轨迹分别如图7和图8所示。

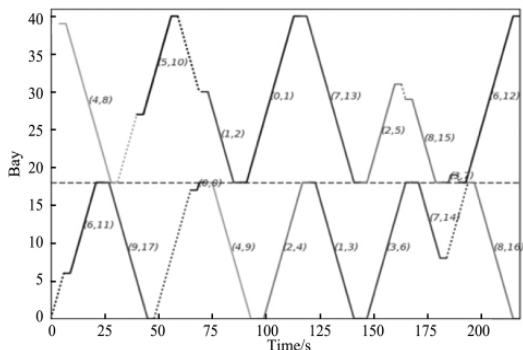


图7 $exB=18$ 时 CPLEX 求解双 ASC 作业轨迹

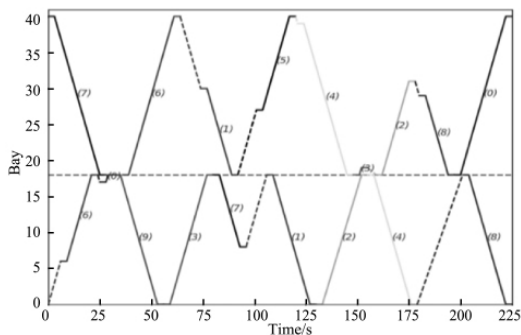


图8 $exB=18$ 时 GA 求解双 ASC 作业轨迹

两种方法均找到了相同的最优接力区位置,生成的 ASC 作业计划不同,且 CPLEX 求得的解更优,避免了双 ASC 干扰问题。但遗传算法也有效降低了干扰的发生,并展示出了更高效的求解能力。如果统一采用表2所示的任务编号方式,则 CPLEX 求得的双 ASC 作业计划为:

$C_1: 6-9-0-14-2-11-3-17-18$

$C_2: 4-5-1-10-7-12-8-13-16$

遗传算法求得的双 ASC 作业计划为:

$C_1: 6-0-9-3-17-11-2-14-18$

$C_2: 7-16-1-5-4-13-12-8-10$

根据上述两种优化方法,只要给定任意的堆场既定作业任务,均可确定出最优的接力区位置,有效解决双 ASC 作业干扰问题,并生成各 ASC 的作业计划,使总完工时间最小。

3.3.3 遗传算法优化分析

当有集装箱船舶靠泊或集疏港相对集中时,自动化集装箱码头往往需要应对大规模的装卸任务。由3.3.1节可知遗传算法是求解该类问题非常有效的方法,因此,对遗传算法的参数设置进行优化也是高效准确求得最优解、快速得到堆场设备调度计划的重要内容。为更直观地反映收敛效果,选择初始任务规模为50的第4组实例进行研究。

当变换迭代次数时,设计50、100、200、300和500五个梯度求解,结果如图9所示。易知当迭代次数取300和500时收敛效果最佳,为提高运算效率,迭代次数取300为宜。同理,通过变换种群规模发现当种群规模为300时收敛效果更佳,结果如图10所示。

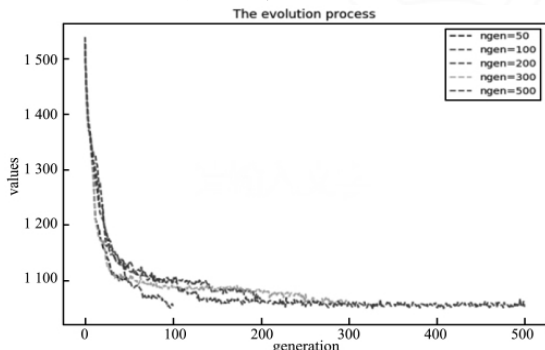


图9 不同迭代次数收敛曲线

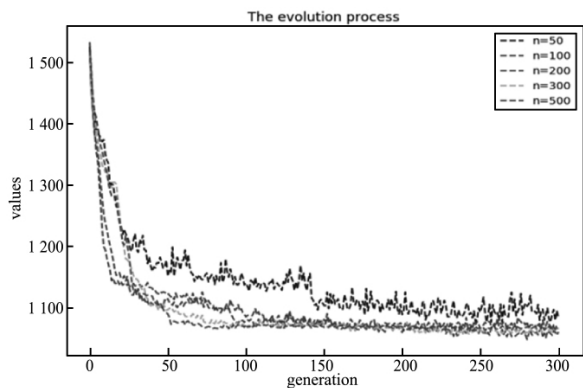


图 10 不同种群规模收敛曲线

对于提高遗传算法求解效率而言,交叉概率 cxb 与变异概率 $mutpb$ 的选取并不是相互无关和独立的,而是存在最优组合问题。针对不同类型的优化问题,所选取的变异率一般随交叉率的增大而呈减小趋势^[15]。因此为提高求解效率,设置几组交叉率—变异率组合如表 4 所示。

表 4 交叉率—变异率组合

组别	1	2	3	4	5	6	7
cxb	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$mutpb$	0.2	0.1	0.05	0.02	0.01	0.005	0.001

取种群规模和迭代次数均为 300,根据表 3 得出各交叉率—变异率组合的收敛图像如图 11 所示。易知当交叉概率为 0.4、变异概率为 0.1 时收敛效果最佳。当接力区位置在箱区中间贝位($exB = 20$)时,在种群规模为 300、交叉概率为 0.4、变异概率为 0.1 条件下迭代 300 次得到完成该 50 个任务的最短完工时间为 1 046 s,求解过程平均历时 241.52 s。经优化,所得结果较表 1 更优,求解时间也大幅缩短,充分论证了该组遗传算法参数优化结果的有效性。该参数配置下收敛曲线如图 12 所示,双 ASC 作业轨迹如图 13 所示。

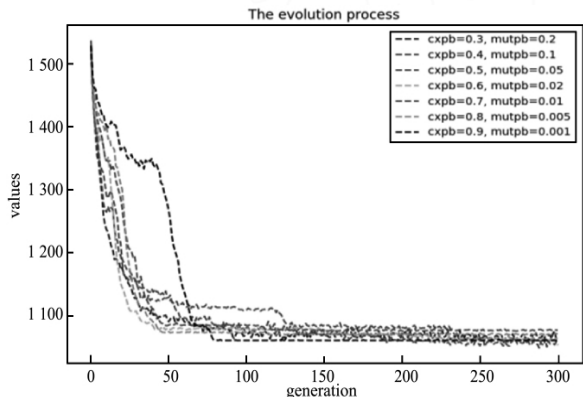


图 11 不同交叉率—变异率组合收敛曲线

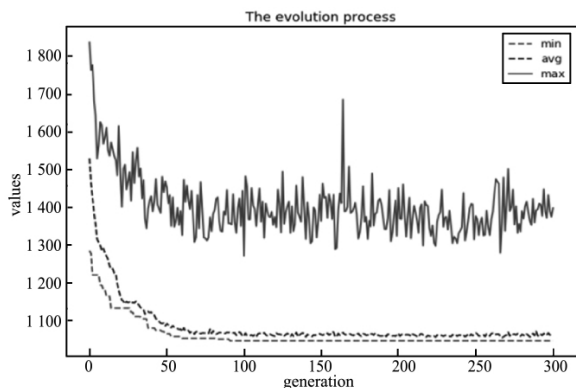


图 12 最优参数配置下收敛曲线

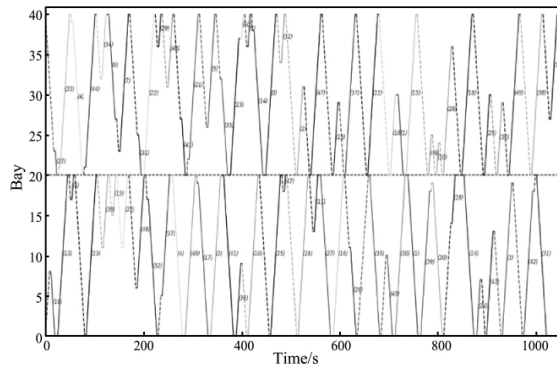


图 13 最优参数配置下双 ASC 作业轨迹图

4 结 语

双轨道吊作业干扰是自动化集装箱码头堆场存取箱作业同时进行的核问题,双 ASC 协同调度优化结果直接影响港口整体运营效率。本文考虑集装箱的堆存和取出两种作业同时进行的情况,设计既定任务下的任务拆分规则,以任务完成总时间最小化为目标,建立双轨道吊协同调度混合整数规划模型,并采用 CPLEX 和遗传算法对多组算例进行求解,主要结论如下:

(1) 通过评估不同规模任务下求解双 ASC 协同调度优化问题的算法性能,得出遗传算法更适合求解大规模设备调度问题。

(2) 变换接力区位置,用 CPLEX 和遗传算法两种方法求解均能找到最优接力区位置,并得到了两台 ASC 各自的作业计划,充分论证了所提出的模型和算法解决自动化集装箱港口双 ASC 作业干扰问题的有效性。

后续研究重点为自动化集装箱码头堆场接力区设计如时间窗条件下动态接力区设计等方面。

参 考 文 献

- [1] Carlo H J, Vis I F A, Roodbergen K J. Seaside operations in container terminals: literature overview, trends, and research directions [J]. European Journal of Operational Research, 2015, 27(2-3): 224-262.

(下转第 143 页)

针对 ARM 架构,除了 NEON 技术可用于优化,还可以考虑利用 GPU 优化,以及利用 ARM 多线程进行相关优化。下一步将着重从 GPU、多线程技术展开进一步的优化工作。

参 考 文 献

- [1] Manjoo F. A murky road ahead for android, despite market dominance [M]. The New York Times, 2015.
- [2] Bradski G, Kaehler A. Learning OpenCV: Computer Vision in C++ with the OpenCV Library [M]. O'Reilly Media, Inc. 2013.
- [3] 龚若皓. 基于嵌入式移动 GPU 的图像编解码并行优化 [D]. 西南交通大学, 2015.
- [4] 陈新泽. 基于 NEON 引擎的 JM 解码器并行优化实现 [D]. 西南交通大学, 2013.
- [5] Wan J, Wang R, Lv H, et al. AVS video decoding acceleration on ARM Cortex-A with NEON [C]//IEEE International Conference on Signal Processing, Communication and Computing. IEEE, 2012: 290-294.
- [6] 龚若皓, 杨斌. Mali T604 GPU 的二维浮点矩阵运算并行优化方法 [J]. 单片机与嵌入式系统应用, 2015, 15(5): 43-46.
- [7] Pujara C, Modi A, Sandeep G, et al. H. 264 Video Decoder Optimization on ARM Cortex-a8 with NEON [C]//2009 Annual IEEE India Conference. IEEE, 2009: 1-4.
- [8] Poudel P, Shirvaikar M. Optimization of Image Processing Algorithms on Mobile Platforms [C]//IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2011, 7871: 787105.
- [9] Du P, Weber R, Luszczek P, et al. From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming [J]. Parallel Computing, 2012, 38(8): 391-407.
- [10] 彭旭康. 基于 ARMv8 架构的 H. 265/HEVC 视频解码优化 [D]. 华中科技大学, 2015.
- [11] Martinez G, Gardner M, Feng W. CU2CL: A CUDA-to-OpenCL Translator for Multi-and Many-core Architectures [C]//Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems. IEEE, 2011: 300-307.
- [12] Jang M, Kim K, Kim K. The Performance Analysis of ARM NEON Technology for Mobile Platforms [C]//Proceedings of the 2011 ACM Symposium on Research in Applied Computation. ACM, 2011: 104-106.
- [13] Jo G, Jeon W J, Jung W, et al. OpenCL Framework for ARM Processors with NEON Support [C]//Proceedings of the 2014 Workshop on Programming models for SIMD/Vector processing. ACM, 2014: 33-40.
- [14] Steenken D, Voß S, Stahlbock R. Container terminal operation and operations research—a classification and literature review [M]//Container Terminals and Automated Transport Systems. Springer Berlin Heidelberg, 2005: 3-49.
- [15] Vis I F A, Roodbergen K J. Scheduling of Container Storage and Retrieval [J]. Operations Research, 2009, 57(2): 456-467.
- [16] Huang Y, Liang C, Yang Y. The optimum route problem by genetic algorithm for loading/unloading of yard crane [J]. Computers & Industrial Engineering, 2009, 56(3): 993-1001.
- [17] Carlo H J, Martínez-Acevedo F L. Priority rules for twin automated stacking cranes that collaborate [J]. Computers & Industrial Engineering, 2015, 89(C): 23-33.
- [18] Briskorn D, Emde S, Boysen N. Cooperative twin-crane scheduling [J]. Discrete Applied Mathematics, 2016, 211: 40-57.
- [19] Jaehn F, Kress D. Scheduling cooperative gantry cranes with seaside and landside jobs [J]. Discrete Applied Mathematics, 2017, 242: 53-68.
- [20] Briskorn D, Angeloudis P. Scheduling co-operating stacking cranes with predetermined container sequences [J]. Discrete Applied Mathematics, 2016, 201: 70-85.
- [21] Vis I F A, Carlo H J. Sequencing Two Cooperating Automated Stacking Cranes in a Container Terminal [J]. Transportation Science, 2016, 44(44): 169-182.
- [22] Hu Z H, 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.
- [23] 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.
- [24] Gharehgozli A H, Vernooij F G, Zaerpour N. A Simulation Study of the Performance of Twin Automated Stacking Cranes at a Seaport Container Terminal [J]. European Journal of Operational Research, 2017, 261(1): 108-128.
- [25] 丁一, 陈杭钦, 林国龙. 成对自动堆垛起重机在集装箱码头中的实时调度研究 [J]. 重庆师范大学学报(自然科学版), 2017, 34(2): 133-141.
- [26] Renner G, Ekárt A. Genetic algorithms in computer aided design [J]. Computer-Aided Design and Applications, 2003, 35(8): 709-726.
- [27] 黎钧琪, 石国桢. 遗传算法交叉率与变异率关系的研究 [J]. 武汉理工大学学报, 2003, 27(1): 97-99.

(上接第 98 页)