

差分进化算法求解自动堆垛起重机调度问题*

王豪¹, 白丹宇², 曾庆成², 侯年华², 曾莎莎²

(1. 大连海事大学 交通运输工程学院; 2. 大连海事大学 航运经济与管理学院, 辽宁 大连 116026)

摘要:【目的】研究了自动化码头双自动堆垛起重机(Automatic stacking crane, ASC)调度问题,为了提高堆场的作业效率,考虑ASC作业过程中的干扰和冲突,以极小化所有作业的总完工时间为目标,建立自动化码头双ASC调度问题的混合整数规划模型。【方法】通过CPLEX求解器对模型进行求解。为了解决中等规模问题,设计了基于任务编码的离散差分进化算法进行求解,并把求解结果与遗传算法求解结果进行对比分析。【结果】所设计的离散差分进化算法能够在要求的时间内解决自动化码头双ASC调度问题;【结论】相较于遗传算法,本文提出的算法寻优能力更强。

关键词: 自动化码头; 双ASC; 差分进化算法; 混合整数规划

中图分类号: O221.6

文献标志码: A

文章编号: 1672-6693(2021)01-0105-09

“一带一路”倡议的提出使得海上集装箱运输量快速增长,港口作为海上运输的重要连接地,作业效率直接影响整个海运供应链的效率。高效的堆场作业能够明显地提高集装箱码头的整体效率和生产能力。为此,一些新建的集装箱码头,如上海洋山4期自动化码头、青岛新前湾自动化码头等均采用自动堆垛起重机(Automatic stacking crane, ASC, 也称作自动化轨道吊)来完成堆场作业。

随着ASC作业系统的应用,以往的堆场作业调度方式已不再适用,针对ASC作业调度问题,国内外学者从不同的角度展开了研究。一些学者分析了设备布置形式和作业模式对堆场作业效率的影响。Gharehgozli等人^[1]研究了堆场箱区内交换区的设置形式对起重机作业性能的影响,并以极小化作业完工时间和因相互干扰引起的起重机等待时间为目标,通过数值仿真,分析了起重机调度方式、集装箱堆存位置以及交换区位置、数量、大小等产生的影响。Speer等人^[2]设计了分支定界算法分析不同的ASC布置形式对堆场作业效率的影响;他们采用近似算法在合理的时间内求得问题的近优解。丁一等人^[3]研究了双ASC的实时调度问题,构建了双ASC调度优化模型,并利用商业优化软件CPLEX进行求解。梁承姬等人^[4]研究了接力模式和混合模式下双轨道吊作业效率问题,以极小化最大完工时间为目标,建立了混合整数规划模型,采用遗传算法对两种作业模式进行分析。

干扰和冲突是影响双ASC作业效率的重要因素,也是学者关注的重点问题。Carlo等人^[5]针对双轨作业系统发生干扰时海侧/陆侧轨道吊优先等级高低的问题,通过仿真实验对14种不同优先等级评定规则下的起重机最大完工时间进行了评估。李敏等人^[6]针对发生冲突的任务,设计了8条优先等级评定规则进行评分,将优先权赋予评分高的任务,并运用Matlab对所建立的优化模型进行求解。周静娴等人^[7]分析了穿越式双ASC执行任务时可能发生的冲突情况,构建了混合整数规划模型优化起重机的作业顺序,并采用商业优化软件Matlab结合数学规划优化器GUROBI进行求解。Nossack等人^[8]研究了自动化堆场内集装箱分配和ASC无冲突调度问题,以极小化最大完工时间与避免双起重机冲突为目标,建立了混合整数规划模型,确定双起重机的任务分配和作业顺序。

目前,国内外对于ASC调度的研究还存在不足:1)双ASC作业过程中干扰和冲突问题仍然是难题;2)一般性地认为海、陆侧I/O点的任务分别由海、陆侧ASC独立完成,即非合作作业模式,并未考虑到一个任务可以由海、陆侧ASC接力完成的情况(合作作业模式)。相较于非合作作业模式,在合作作业模式下,起重机间的干

* 收稿日期: 2020-11-12 网络出版时间: 2021-03-10 16:26

资助项目: 国家自然科学基金面上项目(No. 61873173)

第一作者简介: 王豪,男,研究方向为港口物流系统运营管理优化, E-mail: wanghao_2018@126.com; 通信作者: 白丹宇,男,教授,博士, E-mail: mikebdy@sina.com

网络出版地址: <https://kns.cnki.net/kcms/detail/50.1165.N.20210310.1157.014.html>

扰和冲突只发生在接力区附近,这使得干扰和冲突问题更容易分析;然而,对于起重机的任务排序来说,不仅要考虑接力区附近起重机之间的干扰问题,还要考虑接力任务两个阶段的先后关系,这使得堆场中起重机的调度变得复杂。

本文以极小化 ASC 完成所有任务的总完工时间为目标,考虑任务间的干扰和冲突,构建基于 ASC 合作作业模式的调度优化模型。由于此问题是 NP 困难问题,因此设计离散差分进化算法求解中等规模问题,并通过数值仿真验证算法的有效性。

1 问题描述

不同于传统的堆场布局,配备 ASC 的堆场大多采用垂直岸线式布局,起重机与水平运输工具如集卡、自动导引车(Automated guided vehicle, AGV)等的交接区设置在箱区两端,每个箱区两台起重机同轨运行。

ASC 将来自外集卡及 AGV 的集装箱放入堆场中堆存,并将需要提取的集装箱取出交给 AGV 或者外集卡,或者将堆场中的集装箱重新堆放以满足未来安全、生产的需求^[9]。典型自动化码头布局如图 1 所示。

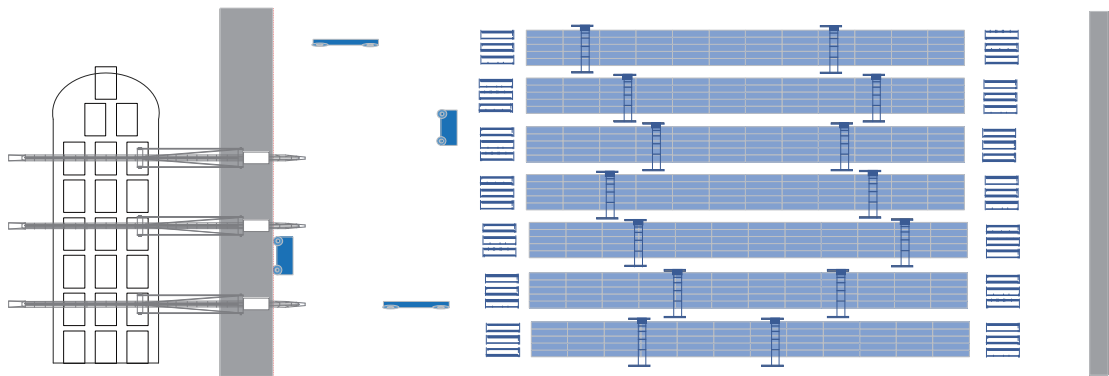


图 1 自动化码头布局图

Fig. 1 Layout of automatic container terminal

在港口的实际生产中,双起重机作业系统通常采用非合作作业模式,在这种模式下,海、陆侧 I/O 点的任务分别由海、陆侧起重机各自完成。该作业模式造成起重机长距离行驶和闲置,导致堆场能耗居高不下。针对此问题,Gharehgozli 等人^[1]提出将堆场中的某一贝位设置为接力区,跨越接力区的任务可以由海、陆侧起重机接力完成,此种作业模式称为合作作业模式。

在采用合作作业模式的双轨作业系统中,对于起点和终点跨越接力区的集装箱运输任务,可以由一个起重机将集装箱从任务起点移动至接力区,然后由另一起重机继续从接力区移动至任务终点。如图 2 所示,对于任务 A,可以先由海侧起重机将集装箱移动至接力区,然后由陆侧起重机从接力区移动至终点位置。

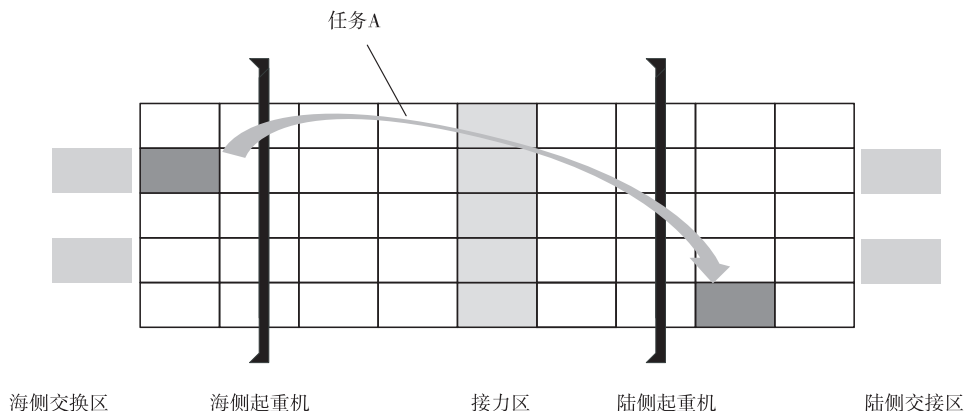


图 2 堆场箱区布局图

Fig. 2 Layout of container yard

2 数学模型

2.1 模型假设

本文研究基于以下假设:

- 1) 只考虑自动化堆场某一箱区情况;
- 2) 集装箱尺寸统一为 20 英尺标准箱;
- 3) 任务的提起位置和释放位置已知,由更高一级的计划层面决策。

2.2 集合和参数

J 表示所有任务的集合, $i \in J = \{1, 2, \dots, m\}$, m 为任务的数量; K 表示起重机的集合, $k \in K = \{1, 2\}$, $k=1$ 表示海侧起重机, $k=2$ 表示陆侧起重机; J_k 表示起重机 k 的任务的集合; J_a 表示不需要接力的任务的集合; J_β 表示需要接力的作业接力部分第一阶段的任务的集合; J_γ 表示需要接力的作业接力部分第二阶段的任务的集合; $t_{i,j}$ 表示起重机从任务 i 的终点至任务 j 的起点的空载行驶时间; $t_{F,j}$ 表示起重机从闲置点至任务 j 的起点的空载行驶时间; $t_{i,T}$ 表示起重机从任务 i 的终点至闲置点的空载行驶时间; s_i 表示起重机作业任务 i 时的重载运行时间; η 表示两个起重机之间的安全距离; δ 表示起重机行驶安全距离 η 花费的时间; b_i^q 任务 i 起点所在的贝位; b_i^l 表示任务 i 终点所在的贝位; Φ_1 表示起点临近的任务对的集合; Φ_2 表示终点临近的任务对的集合; Φ_3 表示任务 i 的起点和任务 j 的终点临近的任务对的集合; Ψ 表示同一任务拆分后的子任务对的集合。

决策变量定义为: c_i 表示起重机完成任务 i 的时间; r_i 表示起重机开始作业任务 i 的时间。当任务 i 和任务 j 为紧前紧后关系且被同一起重机完成时, $x_{i,j} = 1$; 否则 $x_{i,j} = 0$ 。

2.3 混合整数规划模型

$$\begin{aligned} \min \quad & \sum_{i \in J^k} c_i, \\ \text{s. t.} \quad & \sum_{j \in J^k} x_{Fj} = 1, \forall k \in K, \end{aligned} \quad (1)$$

$$\sum_{j \in J^k} x_{jT} = 1, \forall k \in K, \quad (2)$$

$$\sum_{i \in J^k \cup \{F\} \setminus \{j\}} x_{ij} = 1, \forall k \in K, \forall j \in J^k, \quad (3)$$

$$\sum_{i \in J^k \cup \{F\}} x_{il} = \sum_{j \in J^k \cup \{T\}} x_{lj}, \forall k \in K, \forall l \in J^k, \quad (4)$$

$$x_{ij} + x_{ji} \leq 1, \forall k \in K, \forall i, j \in J^k, \quad (5)$$

$$c_i \leq r_i + s_i + M \times \left(1 - \sum_{j \in J^k \cup \{T\}} x_{ij} \right), \forall k \in K, \forall i \in J^k, \quad (6)$$

$$c_i \geq r_i + s_i + M \times \left(\sum_{j \in J^k \cup \{T\}} x_{ij} - 1 \right), \forall k \in K, \forall i \in J^k, \quad (7)$$

$$c_i + t_{ij} - r_j \leq M \times (1 - x_{ij}), \forall k \in K, \forall i, j \in J^k, i \neq j, \quad (8)$$

$$t_{Fj} - r_j \leq M \times (1 - x_{Fj}), \forall k \in K, \forall i, j \in J^k, \quad (9)$$

$$c_i + t_{iT} - y^k \leq M \times (1 - x_{iT}), \forall k \in K, \forall i, j \in J^k, \quad (10)$$

$$c_l + \delta \leq r_j, \forall k, l \in K, k \neq l, (i, j) \in \Psi, \quad (11)$$

$$|r_i - r_j| \geq \delta, \forall k, l \in K, k \neq l, (i, j) \in \Phi_1, \quad (12)$$

$$|c_i - c_j| \geq \delta, \forall k, l \in K, k \neq l, (i, j) \in \Phi_2, \quad (13)$$

$$|r_i - c_j| \geq \delta, \forall k, l \in K, k \neq l, (i, j) \in \Phi_3, \quad (14)$$

$$\Phi_1 = \{(i, j) | i \in J^k, j \in J^l, k \neq l, |b_i^q - b_j^q| \leq \eta\}, \quad (15)$$

$$\Phi_2 = \{(i, j) | i \in J^k, j \in J^l, k \neq l, |b_i^l - b_j^l| \leq \eta\}, \quad (16)$$

$$\Phi_3 = \{(i, j) | i \in J^k, j \in J^l, k \neq l, |b_i^q - b_j^l| \leq \eta\}, \quad (17)$$

$$\Psi = \{(i, j) | i \in J^k, j \in J^l, k \neq l\}. \quad (18)$$

约束(1)和(2)分别为每个起重机选择一个开始任务与一个结束任务;约束(3)保证每个任务只被执行一次;约束(4)规定了每个起重机的作业顺序,且保证每个任务只有一个紧前任务和一个紧后任务;约束(5)检查由同

一起重机执行的两个任务的作业顺序;约束(6)和(7)表明了每个任务的开始作业时间、完成作业时间之间的关系,即起重机完成任务 i 的时间等于开始作业任务 i 的时间加上重载运行时间;约束(8)定义每个任务的开始作业时间,并消除子回路;约束(9)确定了每个起重机开始作业第一个任务的时间;约束(10)确定了每个起重机完成所有任务的时间;约束(11)表明了由同一任务拆分来的任务对 (i, j) 之间的作业时间关系;约束(12)~(14)是对两个起重机可能产生干扰的任务对 (i, j) 的作业时间关系约束;约束(15)~(18)分别定义了集合 $\Phi_1, \Phi_2, \Phi_3, \Psi$ 。

3 求解算法

本文研究的双 ASC 调度问题属于 NP 困难问题^[10],无法在多项式时间内求得最优解;小规模问题可以利用精确算法求得最优解,随着问题规模的增大,求解时间指数增长。在码头生产过程中,起重机调度问题的规模通常较大。本文应用智能优化算法在有限时间内快速求得问题的可行解,以满足实际生产的需要。差分进化(Differential evolution, DE)算法是一种基于种群的全局搜索算法,DE 算法具有收敛速度快、结构简单、鲁棒性强等优点。DE 算法由 Storn 和 Price 于 1995 年首次提出,此后就被广泛地应用在组合优化、生产排程等多种领域。因此,本文设计基于差分进化算法的求解方法解决提出的问题。

3.1 编码

将个体转化为基于任务编号的序列,每个染色体均由 n (任务总数)个基因构成,染色体中以 $1, 2, \dots, n$ 为编号的基因表示的是当前基因位置的任务编号。如给定一个任务总数为 8 的集装箱任务序列,其中 2 号任务需要由两个 ASC 接力完成,编码方式如图 3 所示。

任务编号	2	5	8	1	7	6	4	3
ASC 编号	1, 2	2	1	1	2	1	2	1

图 3 染色体编码示意图

Fig. 3 Coding of chromosome

3.2 初始化

通过随机产生 N 个个体组成初始化种群,并通过正交试验确定能够影响差分进化算法性能的参数。这些参数包括:种群规模 Δ 是进化过程中个体的数目,最大迭代次数 $k_{\max}$ 表示停止条件(即迭代数达到 $k_{\max}$ 后终止算法),变异概率 Z 和交叉概率 Y 。

3.3 变异

在差分进化算法在变异过程中利用两对染色体并行差分去扰动最优个体,如下式所示:

$$V_h^k = X_{\text{best}}^{k-1} \oplus (X_{\alpha_1}^{k-1} - X_{\beta_1}^{k-1}) \otimes Z \oplus (X_{\alpha_2}^{k-1} - X_{\beta_2}^{k-1}) \otimes Z. \quad (19)$$

其中: $V_h^k = [v_{h1}^k, v_{h2}^k, \dots, v_{hn}^k]$ 表示变异个体, X_{best}^k 表示第 k 代种群中的最优个体, $\alpha_1, \beta_1, \alpha_2, \beta_2$ 均是从种群中选出的 4 个各不相同的个体,两两做差分计算。

(19)式中 $\otimes$ 和 $\oplus$ 运算符的运算方法分别为:

$$G_h^k = Z \otimes (X_{\alpha}^{k-1} - X_{\beta}^{k-1}) = \begin{cases} X_{\alpha}^{k-1} - X_{\beta}^{k-1}, & \theta < Z \\ 0, & \text{否则} \end{cases}, \quad (20)$$

$$V_h^k = \text{mod}((X_{\text{best}}^{k-1} + G_{h1}^k + G_{h2}^k + n - 1), n) + 1. \quad (21)$$

其中: G_h^k 就是差分项通式, V_h^k 是变异操作的结果, mod 是取模操作。差分项 G_h^k 在经过减法运算后可能会出现 0、负数甚至超过 n 的值,取模运算的目的是将个体序列中的每一个维度都控制在 $[1, n]$ 范围内。

当变异系数 $Z=0.5$ 时,变异过程如表 1 所示。首先计算出两个个体的差,接着利用变异系数选取差分项中的基因, θ_1, θ_2 是在 $[0, 1]$ 中随机生成的小数,在表 1 中,第一个维度的 -6 因为 θ_1 的值小于变异系数 Z 而保留,而第四个维度的 1 因为 θ_1 的值大于等于变异系数 Z 而置为 0 了。最后利用选取的两个差分项对当下种群中最好的个体进行改进,从表 1 中可以看出在第一个维度 $x_{i,1}$ 上,和为 $(-6) + 4 + 1 = -1 < 0$,经过运算符 $\oplus$ 中的取模操作符之后,该维度的结果被映射到了 $1 \in [1, n]$,对表 1 中的其他维度进行同样的运算过程,最终得到了变异操作后的个体 $V_h^k = [1, 7, 7, 6, 3, 8, 3, 1]$ 。

表 1 变异操作过程($Z=0.5$)
Tab.1 Mutation process($Z=0.5$)

个体	$X_{\alpha 1}$	$X_{\beta 1}$	$X_{\alpha 1}-X_{\beta 1}$	θ_1	G_{h1}^k	$X_{\alpha 2}$	$X_{\beta 2}$	$X_{\alpha 2}-X_{\beta 2}$	θ_2	G_{h2}^k	X_{best}^{k-1}	取模前	V_h^k
$x_{i,1}$	2	8	-6	0.20	-6	7	3	4	0.49	4	1	-1	1
$x_{i,2}$	7	4	3	0.37	3	5	1	4	0.27	4	8	15	7
$x_{i,3}$	5	1	4	0.18	4	1	8	-7	0.59	0	3	7	7
$x_{i,4}$	3	2	1	0.74	0	4	6	-2	0.84	0	6	6	6
$x_{i,5}$	6	3	3	0.42	3	8	5	3	0.33	3	5	11	3
$x_{i,6}$	4	6	-2	0.68	0	3	7	-4	0.13	-4	4	0	8
$x_{i,7}$	8	7	1	0.33	1	2	2	0	0.60	0	2	3	3
$x_{i,8}$	1	5	-4	0.60	0	6	4	2	0.35	2	7	9	1

3.4 交叉

通过将当前个体与变异个体进行交叉操作,得到一个新的个体 U_h^k 。表 2 给出了交叉操作选择基因片段的过程。如表 2 所示,由于 $x_{i,2},x_{i,6},x_{i,8}$ 维度上的随机数 $\theta\geq Y$,所以这 3 个维度的基因片段不能参与之后的交叉操作,最终参与交叉操作的中间个体为 $V_h^{kk}=[1,7,6,3,3]$ 。

表 2 交叉操作选择变异个体的基因片段($Y=0.7$)
Tab.2 Fragment genes of mutation individual ($Y=0.7$)

个体	$x_{i,1}$	$x_{i,1}$	$x_{i,1}$	$x_{i,1}$	$x_{i,1}$	$x_{i,1}$	$x_{i,1}$	$x_{i,1}$
V_h^k	1	7	7	6	3	8	3	1
θ	0.35	0.89	0.33	0.24	0.67	0.91	0.52	0.78
V_h^{kk}	1	7*	7	6	3	8*	3	1*

注:*表示该项不参与之后的交叉操作

图 4 是将变异个体与当前个体的基因片段进行混合的过程。在图 4 中,选择了位置序号是 0,3,6 的 3 个点作为随机插入点,将 V_h^{kk} 分为 3 段(斑点部分)随机地插入到了这 3 个点中,得到一个超出任务数的有重复的基因片段的不可行序列。然后对该序列按照从左到右的顺序进行去重复操作,最终得到了一个可行解 $U_h^k=[5,7,2,8,6,3,1,4]$ 。

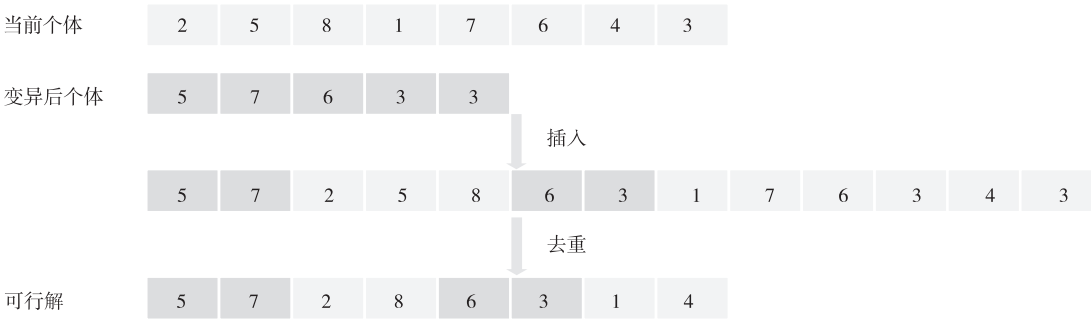


图 4 交叉操作示意图

Fig.4 Crossover operation

3.5 选择

在极小化问题中,通常采用计算目标函数值的方法来选择进入下一代种群的个体。针对本文研究的 ASC 调度问题,给定一个任务序列,根据每个任务对应的集装箱起止位置的属性,可以确定双起重机完成所有任务的总完工时间。具体的计算过程如图 5 所示。

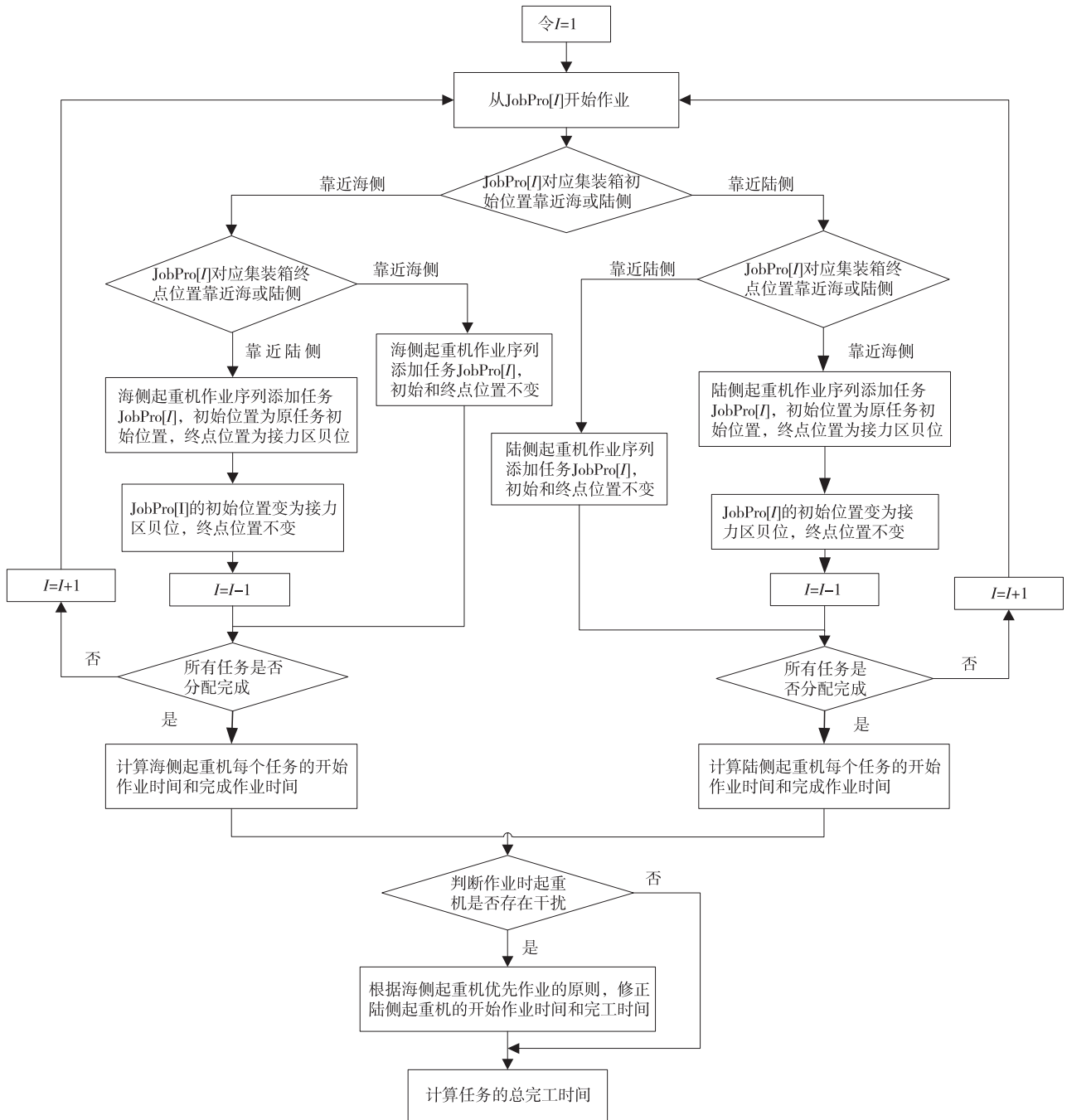


图 5 目标函数计算过程

Fig. 5 Flowchart of objective function calculation

3.6 解码

经过选择操作之后,原始的任务序列被分配给海、陆侧起重机进行操作,海、陆侧起重机作业顺序分别排列出来。例如上面例子中的个体 $[2,5,8,1,7,6,4,3]$ 进行选择操作后,海侧起重机作业序列为 $[2,8,1,6,3]$,陆侧起重机的作业序列为 $[5,2,7,4]$ 。其中任务 2 是由海侧起重机和陆侧起重机接力完成的,海侧起重机的作业顺序为:任务 2→任务 8→任务 1→任务 6→任务 3;陆侧起重机的作业顺序为:任务 5→任务 2→任务 7→任务 4。

4 数值实验

4.1 参数设置

以某自动化码头堆场布局为例进行数值实验,同一规格的集装箱均匀分布。每个箱区有 40 个贝位,0 与 41 号贝位分别表示海侧交换区和陆侧交换区,20 号贝位为接力区。两个起重机不可穿越,起重机匀速行驶,每移动

1 个贝位花费 1 个单位时间,起重机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。

为了保证离散差分进化算法的求解性能,本文通过正交试验对参数设置进行优化,为了平衡解的质量和求解时间,最终确定算法的最大迭代次数为 600,种群规模为 200,交差概率为 0.3,变异概率为 0.1。数值实验在一台处理器为 Inter Core i5-5200U CPU 2.2 GHz、运行内存为 4 GB 的电脑上进行。

4.2 算法有效性分析

为验证本文设计的基于离散差分进化算法的求解方法的有效性,用 C++ 开发算法程序,用 CPLEX 求解混合整数规划模型,并对求解结果进行比较分析。进行数值仿真实验时,针对每种任务规模,随机产生 10 组算例,每组算例运行 5 次,记录实验结果见表 3 和表 4。

表 3 小规模算例结果

Tab. 3 Results of small-scale problems

算例编号	任务数/个	CPLEX 算法		DDE 算法		GAP/%
		函数值 f_1/s	运行时间/s	函数值 f_2/s	运行时间/s	
1	4	248	0.61	248	0.56	0
2	6	362	20.17	364	0.70	0.72
3	8	578	24.08	594	0.82	2.77
4	10	809	139.33	814	1.02	0.66
5	12			1 572	1.37	

注: $GAP = \frac{f_2 - f_1}{f_1} \times 100\%$

表 4 中等规模算例结果

Tab. 4 Results of medium-scale problems

算例编号	任务数/个	最大值/s	最小值/s	平均值/s	平均运行时间/s
6	20	3 210	2 011	2 714.0	2.21
7	50	16 824	12 720	15 113.2	7.71
8	100	66 111	655 380	60 465.6	22.46
9	300	2 010 044	1 735 264	1 935 566.0	155.71
10	600	3 145 491	2 922 725	3 023 269.0	575.57

由表 3 可以发现,当任务数超过 10 时,CPLEX 已经无法在有限时间内(设定 CPLEX 运行时间上限为 7 200 s)求得最优解,而本文设计的差分进化算法,却可以在很短时间内求得近优解,并且所求得的函数值与 CPLEX 求得的最优解之间的误差不大于 2.77%。表 4 展示了离散差分进化算法在求解中等规模问题时的运行时间和目标函数值。由表 4 可以看出,随着问题规模的扩大,离散差分进化算法可以在较短的时间内求出近优解。此外,本文提出的算法完成 300 个任务的排序平均需要 155.71 s,满足自动化码头堆场实际生产过程中对于算法求解性能要求。通过以上对数值实验结果的分析,可以表明,离散差分进化算法在求解不同规模的起重机调度问题时,均可在较短时间内求得近优解;DDE 算法能够有效地解决本文提出的起重机调度问题。

4.3 算法优越性分析

Han 等人^[11]改进了遗传算法,用于有效求解合作作业模式下的双 ASC 调度问题。本文针对所提出的问题,设计实验对改进遗传算法和差分进化算法进行比较分析,遗传算法参数设置参照文献[11]。针对每种问题规模,随机产生 10 组算例,每组算例分别用两种算法运行 5 次,记录每一次运行的结果。之后,逐一对每组算例运行结果进行统计分析,计算相对差异百分比(Relative difference percentage), $R = (f_{\text{ALG}} - f_{\text{OPT}}) \div f_{\text{ALG}} \times 100\%$,其中 f_{ALG} 表示 DDE 及 GA 每次实验的目标函数值, f_{OPT} 是所有 f_{ALG} 中最小的值。

表 5 展示了平均相对差异百分比(R_M)、最小相对差异百分比($R_{\min}$)、最大相对差异百分比($R_{\max}$)及标准差(SD)。由表 5 可以发现,离散差分进化算法的 R_M 明显优于文献[11]中改进的遗传算法的 R_M ,因为针对本文提

出的问题,离散差分进化算法的 R_M 在 0.922 到 2.103 之间,而应用文献[11]中的遗传算法进行求解时, R_M 在 14.417 到 33.992 之间,说明离散差分进化算法的运行结果更接近问题的最优解。此外,DDE 算法的 SD 整体上小于 GA 算法的 SD,说明 DDE 运行结果更稳定,收敛性更强。以上分析表明,针对本文研究的问题,改进的离散差分进化算法求解性能优于改进遗传算法。

表 5 DDE 与 GA 的对比分析
Tab. 5 Results of DDE and GA

算例编号	任务数/个	DDE				GA			
		R_M	$R_{\min}$	$R_{\max}$	SD	R_M	$R_{\min}$	$R_{\max}$	SD
11	50	1.103	0.00	3.193	1.120	38.658	33.992	41.902	2.830
12	100	1.214	0.00	2.390	0.895	32.547	29.196	36.242	2.629
13	200	1.426	0.00	2.558	0.941	23.439	20.656	26.867	2.748
14	300	0.922	0.00	1.869	0.726	17.301	14.417	20.921	2.539
15	500	2.103	0.00	3.557	1.301	17.830	14.741	22.599	3.474

5 结论

集装箱码头堆场作业效率影响着整个码头的物流效率,ASC 调度作为堆场作业的核心,直接影响了港口的整体营运效率。本文研究双 ASC 作业系统合作作业模式下的调度问题,考虑任务的干扰和冲突,以极小化所有任务的总完工时间为目标,建立了重机调度优化模型,通过设计混合差分进化算法求解该问题,并与 CPLEX 及遗传算法求解结果进行对比。为了验证所提出的算法求解中等规模问题的性能,本文设计了 DDE 与 GA 的对比实验。实验结果表明,改进的差分进化算法不仅具有较高的求解效率,所求解的质量更高,而且能够求解中等规模问题,满足自动化堆场作业实际调度的要求。

进一步研究可以在此基础上考虑起重机与水平运输工具(如 AGV、外集卡)协调调度问题,完善起重机调度优化模型,使之更加贴近码头实际情况,具有更强的实用性。此外,堆场中接力区的设计问题也是下一步研究的重点。

参考文献:

- [1] 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.
- [2] SPEER U, FISCHER K. Scheduling of different automated yard crane systems at container terminals[J]. Transportation Science, 2017, 51(1): 305-324.
- [3] 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): 11-19.
DING Y, CHEN H Q, LIN G L. Real-time scheduling of twin automatic stacking crane in container terminal[J]. Journal of Chongqing Normal University (Natural Science), 2017, 34(2): 11-19.
- [4] 梁承姬, 陈维斗, 崔佳诚. 自动化集装箱码头双轨道吊协调调度分析[J]. 计算机应用与软件, 2018, 35(9): 16-21.
LIANG C J, CHEN W D, CUI J C. Analysis of coordinated scheduling of twin-ARMG for automation container terminals[J]. Computer Applications and Software, 2018, 35(9): 16-21.
- [5] CARLO H J, MARTÍNEZ-ACEVEDO F L. Priority rules for twin automated stacking cranes that collaborate[J]. Computers & Industrial Engineering(S0360-8352), 2015, 89: 23-33.
- [6] 李敏, 韩晓龙. 考虑优先权规则的自动化码头穿越式双 ASC 调度[J]. 广西大学学报(自然科学版), 2017, 42(3): 1001-1010.
LI M, HAN X L. Considering priority rules in optimization of passing twin automated stacking cranes in automated terminal[J]. Journal of Guangxi University (Nature Science Edition), 2017, 42(3): 1001-1010.
- [7] 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 J X, HU Z H. Scheduling optimization for cross-over twin automated stacking cranes in automated container terminal[J]. Journal of Computer Applications, 2015, 35(9): 2673-2677.

- [8] 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): 1035-1058.
- [9] 付鹏成, 宋海涛, 李强. 自动化集装箱码头堆场策略[J]. *水运工程*. 2019, 561(10): 32-36.
FU P C, SONG H T, LI Q. Stacking strategy of automated container terminal[J]. *Port & Waterway Engineering*, 2019, 561(10): 32-36.
- [10] GHAREHGOZLI A H, LAPORTE G, YU Y, et al. Scheduling twin yard cranes in a container block[J]. *Transportation Science*, 2015, 49(3): 686-705.
- [11] HAN X L, WANG Q Q, HUANG J W. Scheduling cooperative twin automated stacking cranes in automated container terminals[J]. *Computers & Industrial Engineering*, 2019, 128: 553-558.

Operations Research and Cybernetics

A Differential Evolution Algorithm for Twin ASC Scheduling Problem

WANG Hao¹, BAI Danyu², ZENG Qingcheng², HOU Nianhua², ZENG Shasha²

(1. College of Transportation Engineering, Dalian Maritime University;

2. School of Maritime Economics & Management, Dalian Maritime University, Dalian Liaoning 116026, China)

Abstract: [Purposes] A twin ASC scheduling problem of the automatic container terminal is studied. In order to improve the efficiency of the yard operation, considering the interference and conflicts in the ASC operation process, the integer programming model is established. The optimization goal is to minimize the total completion time. [Methods] The CPLEX solver is used to solve the model. For the medium-scale problems, based on task coding, a discrete differential evolution algorithm is designed. The solution results are compared with those of the genetic algorithm solution. [Findings] The results show that the algorithm proposed can solve the twin-ASC scheduling problem of automatic container terminals within the required time. [Conclusions] The algorithm proposed dominates the genetic algorithm solution.

Keywords: automated container terminal; twin ASC; differential evolution algorithm; mixed-integer programming

(责任编辑 黄 颖)