

考虑碳排放和预约机制的送箱集卡多码头调度问题

范厚明^{1,2}, 郭振峰¹, 李 阳^{1,2}

(1. 大连海事大学 交通运输工程学院, 辽宁 大连 116026; 2. 大连海事大学 战略管理与系统规划研究所, 辽宁 大连 116026)

摘要: 考虑送箱集卡多码头调度的低碳、高效化要求, 及为解决集卡不规律到达导致码头拥挤问题而提出的集卡预约集港机制, 以单一外堆场调用送箱集卡数量最小及多集装箱码头调度方案碳排放总量最小为目标, 构建了单堆场多码头送箱集卡调度模型. 设计了改进的蚁群算法对该模型进行求解, 得到按时间段划分的港外堆场多码头集卡低碳送箱方案, 通过算例验证了所构建模型和算法的有效性. 结果表明, 考虑碳排放和预约机制的送箱集卡多码头调度方案所需集卡数量少, 工作时间短, 兼顾了集装箱码头集卡送箱作业低碳和高效的要求.

关键词: 集卡预约; 碳排放; 集卡调度; 集装箱码头; 蚁群算法

中图分类号: U492.2

文献标志码: A

Truck Scheduling in Delivering Containers from an Outside Yard to Multiple Container Terminals Considering Carbon Emission and the Truck Appointment System

FAN Houming^{1,2}, GUO Zhenfeng¹, LI Yang^{1,2}

(1. Transportation Engineering College, Dalian Maritime University, Dalian 116026, China; 2. Institute of Strategy Management and System Planning, Dalian Maritime University, Dalian 116026, China)

Abstract: In consideration of low carbon emission and efficiency when trucks transport containers to multiple container terminals, and truck appointment system which is taken by container terminals to cope with the problem of irregular arrival of trucks, a scheduling model was established for trucks delivering containers from an outside yard to multiple container terminals, with the objective of minimizing the quantity of the dispatched outside yard's trucks and the total carbon emission of the scheduling plan. A modified ant colony algorithm was designed to solve the model, and the truck scheduling plan for delivering containers from outside container yard to multiple container terminals in different

time periods could be determined. Numerical experiments show that the proposed model and designed algorithm are effective. The study demonstrates that the trucks scheduling scheme can reduce the amount of trucks that the yard must hold and truck drivers' working time. Therefore, the trucks scheduling scheme can balance and meet low-carbon emission and truck efficiency requirements.

Key words: truck appointment; carbon emission; truck scheduling; container terminal; ant colony algorithm

国际贸易量的快速增长和集装箱船舶的大型化趋势, 对集装箱码头的集港效率和服务能力提出了更高的要求. 一方面, 大型集装箱船挂靠的码头作业量和作业强度增大, 另一方面, 大量送箱集卡高峰时间集中在集装箱码头闸口及堆场排队等待作业, 消耗了大量的能源, 排放了大量的 CO₂ 及污染气体, 迫切需要应对和解决. 服务于集港作业的送箱集卡调度一直是集装箱码头集港作业流程的关键环节, 合理优化送箱集卡调度方案对提高车辆资源配置效率、节约能耗、减少碳排放等有直接的影响. 目前有关集卡调度的研究多集中于港口码头内集卡的研究, 对港外集卡进港提送箱作业的相关研究较少. 相对于港口码头内集卡而言, 外集卡服务多个码头且运输距离更远, 码头调度方案对其等待时间的影响更大, 同时由于外集卡资源数量也同样有限, 其等待时间过长势必将降低集卡整体周转效率. 由于大型集装箱船舶对挂靠港口的停泊时间有严格要求, 码头集港时间较短, 且集港量大; 同时集装箱码头往往服务的航线多、挂靠的船舶多, 使得码头作业高峰期有大量集卡在同一时段到达港口进行操作, 容易使码头闸口、堆场和作业线路的内外集卡产生排队、拥堵等诸多问题. 这一方面是对码头及堆场核心资源

收稿日期: 2018-03-26

基金项目: 国家自然科学基金(61473053); 辽宁省社会科学规划基金(L16AGL004); 辽宁省重点研发计划(2018401002)

第一作者: 范厚明(1962—), 男, 教授, 博士生导师, 管理学博士, 主要研究方向为交通运输系统规划与设计、战略管理与系统规划.

E-mail: fhm468@163.com

的浪费,另一方面也增加了集卡在等待过程中的能耗及 CO₂ 排放,提高了运营成本.针对这些问题,一些港口从建立码头预约机制着手,尝试使送箱集卡在可控的规定预约时段内到港,以缓解作业高峰时段码头集卡的拥堵问题,降低堆场码头作业能耗,减少其总碳排放量,提高码头的服务效率和质量.

从 1999 年加拿大温哥华首先实施预约集港以来,长滩、洛杉矶、新加坡和天津等港口引入了集卡预约系统.一些学者针对预约集港及预约机制下送箱集卡调度进行了研究.曾庆成等^[1]利用 Baskett-Chandy-Muntz-Palacios 排队网络优化了每个时段的预约份额,其可以有效地减少集卡在码头的周转时间,较好地处理到达过程不平稳的排队问题.Namboothiria 等^[2]以降低车队运输成本为目标构建了集卡调度模型,研究集卡预约系统对提高周转效率的影响.蒋美仙等^[3]提出了基于 Poisson 过程的港口码头集卡预约调度模型,根据到港船舶船期和预出口集装箱数目,模型对每艘到港船舶进行了预约时间段和各时间段计划预约集装箱数量分配.Huynh 等^[4]模拟了集卡在堆场的作业过程,利用数学模型求解了每个时间段最多可接受的预约份额.邵乾虔等^[5]构建了码头集疏港联动预约模式,以时间残值矩阵为核心的线性仿真数学模型验证了联动预约模式的有效性.许巧莉等^[6]研究了非平稳到达情况下的集卡预约优化问题.Guan 等^[7]基于 $M/E_k/c$ 排队模型(顾客到达间隔时间分布为 M 负指数分布, E_k 表示服务时间分布为 k 阶爱尔兰分布,服务台数目为 c)建立了集卡预约优化模型.Chen 等^[8]建立了预约机制下集卡服务非稳态排队模型,在此基础上设计了包括集卡到达优化、拥堵收费设计的两阶段码头外部集卡管理方法.此外,曾庆成等^[9]研究了集装箱码头装卸过程中集卡调度问题,建立了集卡调度动态模型,应用 Q 学习算法(Q-learning algorithm,一种强化学习算法)进行了求解.丁一等^[10]以桥吊调度计划为依据,通过给定轮胎吊和集卡关联任务集与作业要求时间,研究了轮胎吊-集卡的调度问题特性.Phan 等^[11]、Zhang 等^[12]及 Chen 等^[13]对具有能源约束、集卡到达时间、时间窗限制的集装箱码头集卡调度优化问题进行了分析求解.综上,目前的相关研究多是对单一外堆场与单一码头间的集卡调度问题,而现实中的港外堆场服务能力较强,一般同时为多个码头执行集港服务.本文在考虑堆场、码头节能减排发展要求的基础上,从碳排放和预约机制的视角出发,对单一港外堆场服务多

码头的送箱集卡调度问题进行研究,通过合理安排堆场集卡的送箱作业顺序,降低调用集卡数量,提高集卡作业效率,减少其在码头的作业等待时间,从而降低集卡多码头调度作业 CO₂ 总排放量,提升堆场与码头的协作水平.

1 送箱集卡调度模型

1.1 问题描述

在传统集装箱码头针对远洋运输的集港业务流程中,船方会在船舶到港前一段时间(如前 96 h)向码头预报船期,并在船舶到港前 24 h 还会有船期确报,集装箱码头根据船期以及码头的出口箱量、码头闸口和场内机械的作业效率确定每条船舶的集港时间.港外堆场根据各码头公布的各集装箱对应船舶的集港时间、对应码头的集港箱量大小、集港时间紧迫程度等信息,考虑自身的运力配置情况,制定集卡调度方案将集装箱运抵集装箱码头.

由于船舶集中到达、码头场桥和岸桥等核心调度资源能力不足、码头闸口拥挤等原因,这种传统的集卡调度方案容易产生集卡资源过度占用,在码头闸口、港内堆场等待时间过长等问题,因此传统集卡调度在制定方案时需要堆场保持一定数量的集卡来应付各种突发需求.由于堆场集卡保有量较高,集卡在作业过程中怠速等待时间过长,导致车辆能耗及 CO₂ 总排放量较高,堆场运营成本较高.

依托现代信息和网络通信技术有效建立码头和堆场之间的联系.集装箱码头根据船期以及码头的出口箱量、码头闸口和场内机械的作业效率确定每条船舶的集港时间.码头将集港时间平均分成若干时间间隔相同的预约时段.堆场根据自身运力情况选择合适的预约时段,并给出该时段的集港箱量.码头经过平衡确定最终的集港计划,并通知各个堆场,堆场按照具体的集港计划来安排自身运力并进行集港工作.

图 1 显示了单一港外堆场向多个集装箱码头的送箱集卡作业流程.在码头预约机制下,为了让集卡可以在其所预约时段到达闸口,码头需要保证一定长度的预约时段.码头一般将 24 h 划分为 12 个预约时段(每时段 2 h),每个时段码头给各个堆场一定的预约份额,堆场可根据各个码头的集港计划以及自身实时运力情况选择预约时段.码头根据各堆场所提交的预约时段及相应预约箱量确定最终的堆场集港计划.堆场按照最终的码头集港计划制定集卡调

度计划方案.

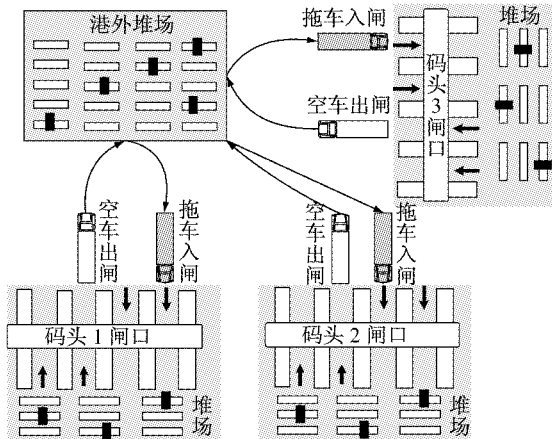


图 1 单港外堆场对多码头的送箱集卡作业流程示意图

Fig. 1 Operation flow of delivery trucks from single outside yard to multi-terminal

由于码头在预约时段内的集卡预约份额是有限的,码头不允许堆场在全天 12 个时段均申请集港作业.在预约机制下各码头提前通知堆场其船舶集港计划(一般提前 24 h),码头会按照计划为堆场分配相应的送箱时间窗(即可接受其集港的预约时段区间).堆场与码头最终确定集港作业计划.一般船舶在本港的装箱量决定了码头的集港箱量,不同码头船舶的出口箱集港时间对堆场而言可能在预约时段上有所重叠,在该时间窗内堆场运抵码头的集装箱数量必须满足码头集港计划要求.因此本文研究的是在堆场现有集卡资源基础上,在码头预约机制下,单一港外堆场向多集装箱码头的送箱集卡调度的优化问题,满足码头预约时段有限集卡预约份额要求,使堆场最终调用的集卡数量最少的同时节约能耗,减少集卡调度方案中 CO₂ 的总排放量,提高堆场的运营效率和效益.

1.2 问题假设

(1) 预约机制下码头将每天划分为 12 个预约时段(每时段 2h),各码头时段划分一致.

(2) 预约机制下堆场所调用集卡从 1 时段 0 时刻开始计时,其作业时间主要包括装卸箱时间、运输时间、等待时间.在堆场的装箱时间主要由其场桥决定,一般取平均值;码头内卸箱时间及运输时间可统一为码头的集卡操作时间,一般取平均值,各码头操作时间各异;运输时间与堆场和码头间的距离直接相关;等待时间包括堆场等待时间、码头等待时间及闸口排队时间,堆场及码头的等待时间与集卡调度方案有关,码头闸口排队等待时间与码头相关,各码头一般根据其情况取平均值.

(3) 假设多集装箱码头所需要的总箱量一定,集卡从单一堆场对多码头进行送箱作业时其总的路程是不变的,但不同优化方案的码头作业顺序会对堆场集卡调用数量以及由于怠速等待运行造成的碳排放量有直接的影响.

(4) 集卡从堆场出发时仅能装载一个集装箱,在相应码头完成卸箱操作后应返回始发堆场等待继续调度,集卡在码头等待及装卸箱作业时怠速运行,在堆场等待调度时无能耗.

(5) 堆场集卡型号相同,行驶速度、油耗等性能相同,可在一个预约时段内多次往返堆场码头完成集港任务.

(6) 在预约机制下,集卡能准时到达集港作业计划码头,不考虑其他干扰因素对集卡的影响.

(7) 预约机制下,码头会为堆场分配集港时间窗,时间窗所包括的预约时段和预约份额由码头给定,码头堆场共同制定集港作业计划后,堆场出口集装箱必须在码头集港时间窗内到达港口.

1.3 模型建立

目标函数分为两部分, K_0 表示堆场所调用集卡数量, E 表示相应集卡调度方案的总 CO₂ 排放量.目标函数如下:

$$\min \begin{cases} K_0 = \max \left\{ \sum_{k \in K} y_{i,k} \right\} \\ E = \left(\sum_{i \in I} \sum_{j \in J} \sum_{n \in N} \sum_{k \in K} y_{ijn,k} (t_{j2} + t_{j3} + t_{ijn2,k}) \cdot c_u + D_j (c_M + c_L) \right) e_{CO_2} \end{cases} \quad (1)$$

式中: i 为预约时段,将 24 h 决策期等分为 I 个预约时段, $i = 1, 2, \dots, I$; j 为集装箱码头编号, $j = 1, 2, \dots, J$; n 为集卡在某一时段的第 n 次送箱, $n = 1, 2, \dots, N$; k 为堆场集卡编号, $k = 1, 2, \dots, K$; $y_{i,k}$ 为决策变量,表示集卡 k 在预约时段 i 是否参与过送箱,若是则其值为 1,否则为 0; $y_{ijn,k}$ 为决策变量,若集卡 k 在 i 时段的第 n 次送箱至码头 j ,则 $y_{ijn,k} = 1$,否则 $y_{ijn,k} = 0$; t_{j2} 为集卡在码头 j 闸口的平均等待时间; t_{j3} 为集卡在码头 j 的平均操作时间; $t_{ijn2,k}$ 为集卡 k 在 i 时段第 n 次送箱到达码头 j 后,在码头的等待时间; c_u 为集卡的怠速油耗, $L \cdot \text{min}^{-1}$; D_j 为堆场到码头 j 的距离, km ; c_M 为集卡重载油耗,为 $L \cdot \text{km}^{-1}$; c_L 为集卡轻载油耗, $L \cdot \text{km}^{-1}$; e_{CO_2} 为燃油 CO₂ 排放系数, $\text{kg} \cdot L^{-1}$.

目标函数计算时应首先确定各时段堆场调用的集卡数量,再从中取最大值作为堆场总集卡调用量,

取其最大值堆场能保证每个时段各个码头的送箱要求都能得到满足,且未调用集卡可在堆场熄火等待,减少碳排放.其相应集卡调度方案总的碳排放量与集卡调度方案中集卡行驶距离、作业及怠速等待时间直接相关,目标函数优化目标是使两个目标都尽可能达到最优,保证最终的优化调度方案较少占用堆场集卡,减少总碳排放量.

约束条件表示如下:

(1) 数量约束

$$\sum_{i \in I} x_{ij} = Q_j, \forall j = 1, 2, \dots, J \quad (2)$$

$$x_{ij} = \sum_{k \in K} \sum_{n \in N} y_{ijn,k}, \forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J \quad (3)$$

$$x_{ij} \leq h_{ij}, \forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J \quad (4)$$

$$y_{ijn,k} \in \{0, 1\}, \forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J, \forall n = 1, 2, \dots, N, \forall k = 1, 2, \dots, K \quad (5)$$

$$y_{i,k} \in \{0, 1\}, \forall i = 1, 2, \dots, I, \forall k = 1, 2, \dots, K \quad (6)$$

式(2)~(6)中: x_{ij} 为堆场在预约时段*i*送达码头*j*的箱量; Q_j 为码头*j*需要的集港总箱量; h_{ij} 为预约时段*i*码头*j*允许堆场运送的最大送箱量.

式(2)表示在码头*j*的预约送箱时间窗内,堆场各时段总的送箱量应满足码头*j*的要求.式(3)表示*i*预约时段下从堆场送往码头*j*的总箱量.式(4)表示预约时段*i*内堆场到码头*j*的送箱量要满足其在该时段内最大送箱量的约束.式(5)、(6)为决策变量属性.

(2) 时间约束

$$\begin{aligned} T_{ijn,k} &= T_{idn,k} + t_{j1} \\ &\forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J, \\ &\forall n = 1, 2, \dots, N, \forall k = 1, 2, \dots, K \end{aligned} \quad (7)$$

$$\begin{aligned} T_{in,k} &= T_{ijn,k} + t_{j2} + t_{j3} + t_{j1} \\ &\forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J, \\ &\forall n = 1, 2, \dots, N, \forall k = 1, 2, \dots, K \end{aligned} \quad (8)$$

$$\begin{aligned} T_{id1,k} &= T_{(i-1)m,k} + t_{ij11,k} + t_0 \\ &\forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J, \\ &\forall n = 1, 2, \dots, N, \forall k = 1, 2, \dots, K \end{aligned} \quad (9)$$

$$\begin{aligned} T_{idn,k} &= T_{in,k} + t_{ij1,k} + t_0 \\ &\forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J, \\ &\forall n = 1, 2, \dots, N, \forall k = 1, 2, \dots, K \end{aligned} \quad (10)$$

$$\begin{aligned} T_s + (i-1)L &\leq T_{ijn,k} \leq T_s + iL \\ &\forall i = 1, 2, \dots, I, \forall j = 1, 2, \dots, J, \\ &\forall n = 1, 2, \dots, N, \forall k = 1, 2, \dots, K \end{aligned} \quad (11)$$

式(7)~(11)中: $T_{ijn,k}$ 为集卡*k*在*i*时段第*n*次送箱

至码头*j*的时刻,若 $T_{ijn,k}$ 处于时段*i*,则 $T_{idn,k}, T_{in,k}$ 计入时段*i*而不考虑其实际所处时段; $T_{idn,k}$ 为集卡*k*在*i*时段第*n*次离开堆场的时刻; $T_{in,k}$ 为集卡*k*在*i*时段第*n*次返回堆场的时刻; t_{j1} 为集卡从堆场到码头*j*的行驶时间; t_0 为集卡在堆场的装箱时间, min; L 为每个预约时段长度, min; T_s 为第一个预约时段开始时刻; $t_{ij1,k}$ 为集卡*k*在*i*时段第*n*次送箱至码头*j*前,在堆场的等待时间.

式(7)表示集卡*k*第*i*时段第*n*次送箱到达码头*j*的时刻.式(8)表示集卡*k*第*i*时段第*n*次回到堆场的时刻.式(9)表示集卡*k*在第*i*时段第1次送箱离开堆场的时刻,式中 $T_{(i-1)m,k}$ 表示集卡*k*在第*i*-1时段最后一次返回堆场的时刻, $t_{ij11,k}$ 表示集卡*k*第*i*时段第1次离开堆场前在堆场内的等待时间.式(10)表示集卡*k*第*i*时段第*n*($n \geq 2$)次离开堆场送箱的时刻,式中 $T_{in,k}$ 表示集卡*k*第*i*时段第*n*-1次送箱返回堆场的时刻.式(11)保证在预约时段*i*送往码头*j*的集装箱要满足其预约时段*i*的时间限制.模型中相关的数量及时间约束能保证堆场集卡在预约时间段内满足各个码头的送箱要求.

2 算法设计

本文设计了改进的蚁群算法,采用人工蚂蚁来模拟堆场送箱集卡进行调度优化^[14].在每次循环中,每只蚂蚁分别模拟*K*辆集卡,*K*即为本次循环中堆场所调用的集卡数量.在所设计算法中,状态转移规则和信息素更新规则是集卡调度模拟的2个重要的规则,直接决定了蚂蚁的搜索状态,结合集卡调度特点,算法对其做了相应改进,以提高算法对最优方案的搜索效率,保证集卡作业顺序的合理性.

2.1 解的结构设计

针对所构建集卡调度模型的特点,本文首先在基本蚁群算法中加入时间窗和分割配送的设计^[15].时间窗设置能使蚂蚁有效地判别码头不同预约时段内的送箱需求,进而合理安排最适合的集卡执行相应配送.与车辆路径问题及旅行商问题等不同,在集卡调度问题中车辆往往需要在堆场和各集装箱码头间执行多行程反复配送,因此结合其多行程的调度特点,算法引入了分割配送设计,当蚂蚁在时间窗和运量约束下完成第1辆集卡调度后,从其开始的预约时段对第2辆集卡进行模拟直到所有任务都被完成,以每辆集卡的路径串序列代表部分解方案,所有蚂蚁所模拟集卡路径及时间过程即为完整的求解方案.

改进的蚁群算法初始解路径串由集卡编号、预约时段、送箱序数、送箱码头、前一次送箱完成后返回堆场时刻、堆场等待时间等元素组成,例如(1,1,1,3,0,0)→(1,2,1,1,15,5)→(1,2,2,4,30,0)表示 1 号集卡的全部调度过程,集卡从 0 开始,在等待 0 后进行第 1 时段的第 1 次送箱,送箱目标为码头 3;返回堆场的时刻为 15,等待 5 后执行第 2 时段的第 1 次送箱,送箱目标为码头 1;再次返回堆场的时刻为 30,等待 0 后进行第 2 时段的第 2 次送箱,送箱目标为码头 4. 集卡全部路径串即组成了港外堆场在多集装箱码头间的总调度方案.

2.2 信息素更新规则

信息素更新规则主要包括集卡行驶后信息素更新时间及更新量. 根据模型的特点,算法设定在所有蚂蚁均都遍历一次循环后更新信息素,对每只蚂蚁 m ,更新规则如下:

$$\tau_{ijn,k}(t+1) = (1-\rho)\tau_{ijn,k}(t) + \Delta\tau(t)y_{ijn,k} \quad (12)$$

式中: $\tau_{ijn,k}(t)$ 表示第 t 次循环集卡 k 在 i 时段第 n 次送箱至码头 j 的信息素浓度. 当蚂蚁完成了一次循环后,路径上的信息素会相应挥发, ρ 表示信息素挥发系数. $\Delta\tau(t)$ 表示第 t 次循环时所选定路径的信息素增加量,未选定路径信息素浓度在挥发后不相应增加,考虑到调度方案中集卡数量的重要性,算法将当前方案集卡调用数量引入信息素增加量计算,

$$\Delta\tau(t) = \frac{Q}{(K_m - K_n)^2}, \text{ 其中 } Q \text{ 表示信息素增加常量, } K_m \text{ 表示第 } t \text{ 次循环蚂蚁 } m \text{ 模拟的集卡数量, } K_n \text{ 为常量,根据调度任务其取值小于 } K_m. \text{ 为了突出最优集卡调度方案,对当前最优解中路径信息素进行一次额外的更新,其大小为 } \Delta\tau(t) = \frac{2Q}{(K_m - K_n)^2}.$$

2.3 状态转移规则

算法在求解过程蚂蚁会根据道路上遗留的信息素按照概率来进行下一路段路径的选择,信息素浓度越高,路径被选中的概率越大. 因此需要设定状态转移规则来计算路径被选中的概率,完成集卡路径的构造. 在蚁群算法中,蚂蚁对于路径的选择主要是由信息素浓度和启发函数决定的,状态转移规则如式(13)所示.

$$p_{ijn,k}(t) = \begin{cases} \frac{\tau_{ijn,k}^\alpha(t)\eta_{ijn,k}^\beta(t)}{\sum_{j \in J_{m,k}} \tau_{ijn,k}^\alpha(t)\eta_{ijn,k}^\beta(t)}, & j \in J_{m,k} \\ 0, & j \notin J_{m,k} \end{cases} \quad (13)$$

式中: $p_{ijn,k}(t)$ 表示第 t 次循环时集卡 k 在第 i 时段

第 n 次前往码头 j 的概率; $\tau_{ijn,k}(t)$ 表示第 t 次循环时集卡 k 在第 i 时段第 n 次送箱前往码头 j 的路径上的信息素浓度; $\eta_{ijn,k}(t)$ 表示第 t 次循环时集卡 k 在第 i 时段第 n 次送箱前往码头 j 的启发函数; α, β 分别表示集卡调度过程中信息素浓度和启发函数的相对重要度; $J_{m,k}$ 表示集卡 k 第 i 时段第 n 次送箱所允许前往的码头集合.

2.3.1 启发函数的构造

在基本蚁群算法中,启发函数主要与路段的路径长度有关,路径越短其被选中概率越高,由于本问题集卡送箱量是固定的,其送箱与否与路径长度无关,因此在构建启发函数时主要考虑集卡在堆场的等待时间,保证集卡能在各码头间高效周转. 集卡相应的启发函数如式(14)、(15)所示.

$$\eta_{ijn,k}(t) = w_{ijn,k} \quad (14)$$

$$w_{ijn,k} = 1 - \frac{t_{ijn1,k}}{\sum_{j \in J_{m,k}} t_{ijn1,k}} + \epsilon, \forall j \in J_{m,k} \quad (15)$$

式(14)、(15)中: $w_{ijn,k}$ 表示集卡在选择下一前往的码头时能否被选中的概率,与其在堆场的等待时间 $t_{ijn1,k}$ 直接相关,以有等待时间的集卡 1 和无等待时间的集卡 2 为例,由于 $t_{ijn1,1} > 0$ 使 $w_{ijn,1} < 1, t_{ijn1,2} = 0$ 使 $w_{ijn,2} = 1$, 因此集卡 2 被选中概率大于集卡 1, $t_{ijn1,k}$ 越大,集卡被选中前往的概率越小; ϵ 为一极小的正数; $J_{m,k}$ 表示蚂蚁 k 在第 i 时段第 n 次送箱时能到达,且满足时间窗及最大送箱量约束的码头集合.

2.3.2 禁忌表设计

在满足堆场码头时间窗、预约时段码头最大预约份额以及码头总集港箱量约束下,集卡调度问题并不限制每辆集卡访问某码头的次数. 码头禁忌表设计如下: 设蚂蚁 k 在堆场空闲,可以接受任务. 若 $J_{m,k} = \emptyset$, 则表明集卡 k 在第 i 时段的送箱任务全部完成; $J_{m,k} \neq \emptyset$, 则计算码头集合 J 中的转移概率 $p_{ijn,k}(t)$.

2.4 算法步骤

改进的蚁群算法流程如图 2 所示.

算法步骤如下:

Step 1 输入码头初始信息 J, I , 码头送箱时间窗 $[T_{j,s}, T_{j,f}]$, 其中 $T_{j,s}$ 和 $T_{j,f}$ 分别表示码头开始接箱时间和码头终止接箱时间, Q, h_{ij} , 各时间信息 $t_0, t_{j1}, t_{j2}, t_{j3}$, 蚂蚁数量 M , 堆场集卡总量 K ; 设置初始信息素浓度 $\tau_{ijn,k}(0), Q, \rho$, 相对重要度 α, β , 最大迭代次数 N ; 令当前循环次数 $t=1$.

Step 2 令当前集卡数量 $k_m=1, i=1, n=1, x_{ij}=0, q_j=0, T_{1,0,k}$ 设置为第 1 预约时段开始时刻 (q_j 表

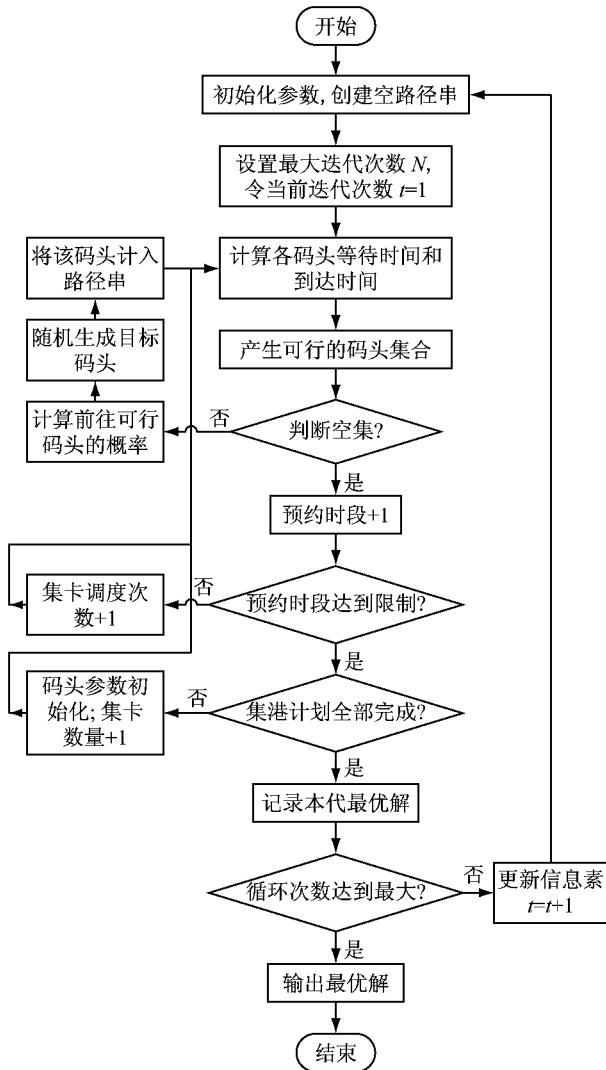


图 2 改进蚁群算法流程图

Fig. 2 Modified ant colony algorithm procedure

示码头 j 已完成的送箱量, k_m 表示蚂蚁 m 模拟的集卡数量).

Step 3 计算集卡 k 在第 i 时段第 n 次前往码头 j 之前, 在堆场的等待时间 $t_{ijn,1,k}$ 和到达码头 j 的时刻 $T_{ijn,k}$ (假设每辆集卡都希望尽早到达码头), 判定其可行性, 即码头是否有集港需求, 堆场在第 i 时段最大送箱量是否饱和, 是否可以在时段 i 到达, 是否有时间窗可以响应. 遍历结束后, 产生可行的码头集合 $J_{in,k}$. 若 $J_{in,k} = \emptyset$, 则转入 Step 5; 若 $J_{in,k} \neq \emptyset$, 则转入 Step 4.

Step 4 对 $J_{in,k}$ 中的码头, 按式(13)计算第 t 循环蚂蚁 m 在第 i 时段第 n 次前往的码头 j 的概率, 决定本次前往的码头, 计算返回堆场的时刻 $T_{im,k}$, 将该元素加入路径串序列. 令 $y_{ijn,k} = 1, n = n + 1, x_{ij} = x_{ij} + 1, q_j = q_j + 1$, 记录蚂蚁 m 模拟集卡调度的解, 返回 Step 3.

Step 5 $i = i + 1$. 若 $i \leq 12$, 则令 $n = 1$, 返回 Step 3. 若 $i > 12$, 判断所有码头的集港需求是否得到满足, 若是则 $k_{best} = \min \{k_m\}$, 记录最优解, 转 Step 6; 若不是, 则 $k_m = k_m + 1, i = 1, n = 1, T_{1r0,k} = T_{1,s}$, 返回 Step 3.

Step 6 若 $t < N$, 则根据此循环中蚂蚁 m 路径信息更新信息素, $t = t + 1$ 转 Step 2; 若 $t \geq N$, 则输出历代得到的最优结果, 从中选取最优解.

3 算例分析

3.1 算例描述

某堆场拥有 100 辆可调度的集卡, 堆场与港口的 7 个码头建立了相应的预约机制. 码头将一天 24 h 划分为 12 个预约时段, 每个预约时段长度为 2 h. 集卡平均行驶速度为 $60 \text{ km} \cdot \text{h}^{-1}$, 其重载油耗为 $1.2 \text{ L} \cdot \text{km}^{-1}$, 轻载油耗为 $0.8 \text{ L} \cdot \text{km}^{-1}$, 怠速状态下油耗为 $2.5 \text{ L} \cdot \text{h}^{-1}$, 燃油的 CO_2 排放系数为 $2.65 \text{ kg} \cdot \text{L}^{-1}$, 集卡在堆场的平均装箱时间 t_0 为 3 min. 堆场与各码头间的距离 l_j , 码头集港箱量 Q_j , 码头送箱时间窗信息 $T_{j,s}, T_{j,t}$, 集卡在各码头闸口的平均等待时间 t_{j2} , 在码头平均操作时间 t_{j3} 等信息如表 1 所示, 不同预约时段码头对堆场的最大预约份额如表 2 所示. 为了便于计算, 案例中的时间单位在计算时都用分钟表示.

表 1 各码头集港信息

Tab. 1 Dependent information of terminals

j	l_j/km	$Q_j/\text{箱}$	$T_{j,s}/\text{min}$	$T_{j,t}/\text{min}$	t_{j2}/min	t_{j3}/min
1	50	78	120	840	6	10
2	40	51	0	480	4	7
3	40	44	240	720	7	4
4	30	56	480	1 080	5	4
5	20	39	360	840	6	5
6	30	30	720	1 080	6	9
7	40	61	600	1 320	7	5

3.2 算例求解

为检验本文调度模型及算法的适用性, 算法采用 MATLAB2012 进行模型求解, 在奔腾双核处理器 E5400, 主频 2.7 GHz, 内存为 3.00 GB 的计算机上运行. 为了选择较好的算法参数组合, 令 α, β, ρ 在 $[1, 2], [1, 3], [0, 1]$ 中取值^[16], 通过对算例进行大量实验验证, 确定所设计改进蚁群算法的各项参数取值如下: 信息素初始浓度设置为 1, $K_n = 30$, 信息素挥发系数 $\rho = 0.5$, 相对重要度 $\alpha = 1$, 相对重要度 $\beta = 2$, 蚂蚁数量 $M = 30$, 信息素增加常量 $Q = 100$. 当迭代次数设为 500 时, 算法可在迭代次数内找到较

表 2 各码头预约时段允许堆场的最大送箱量

Tab.2 Maximum delivery volume of container allowed in each terminal at appointment periods 箱

i	j						
	1	2	3	4	5	6	7
1	15	14	17	16	14	17	9
2	17	10	8	12	11	12	17
3	9	14	12	17	13	12	17
4	13	14	9	9	12	11	13
5	12	15	17	10	8	14	8
6	8	12	8	9	10	11	10
7	11	8	15	9	9	9	11
8	9	10	16	16	9	12	16
9	15	17	16	13	10	11	8
10	11	9	8	13	12	10	8
11	13	16	11	9	8	12	9
12	9	13	10	16	17	8	14

优的解,调度结果如图 3 所示.

由图 3 可知,随着迭代次数的增加,最优调度方案中集卡数量渐渐趋于稳定,当算法迭代到 400 代时,所调用的集卡数量波动越来越小,目标值趋于收敛.此时的送箱集卡最优调度方案所使用的集卡数

量为 34,最优方案中集卡调度顺序及其作业情况如表 3 所示.在各集卡的路径方案中,由 1×5 的数组来标记其送箱及作业情况.数组中的 5 个数字依次为该集卡本次送箱的预约时段、本时段的送箱序号、送箱码头序号、前一次送箱完成后返回堆场的时刻、本次送箱在堆场的等待时间.表 4 中数据给出了表 3 各集卡及最终调度方案中运输里程、作业时间及 CO₂ 排放量等信息.

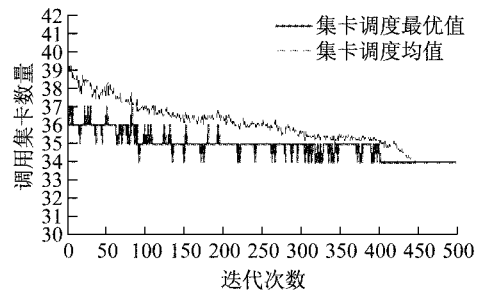


图 3 算法迭代 500 次优化结果

Fig. 3 Optimized results of algorithm at iteration 500

表 3 预约机制下算法求解最优集卡调度方案

Tab. 3 Optimal scheduling plan of truck in appointment system

集卡编号	送箱服务次序																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	(1, 1, 2, 0, 0)	(2, 1, 2, 94, 2, 188, 0)	(2, 2, 2, 188, 1, 282, 0)	(3, 1, 1, 282, 3, 401, 0)	(4, 1, 1, 495, 5, 614, 0)	(5, 1, 5, 668, 5, 722, 0)	(6, 1, 5, 614, 5, 668, 0)	(6, 2, 5, 722, 5, 776, 0)	(7, 1, 5, 776, 4, 830, 0)	(7, 2, 4, 830, 902, 0)	(8, 1, 4, 974, 0)	(8, 2, 4, 974, 0)	(9, 1, 4, 1046, 0)	(9, 2, 4, 1046, 0)	(10, 1, 7, 1118, 0)	(11, 1, 7, 1213, 0)	
2	(1, 1, 2, 0, 0)	(2, 1, 2, 94, 2, 188, 0)	(2, 2, 2, 188, 2, 282, 0)	(3, 1, 1, 282, 1, 376, 0)	(4, 1, 1, 495, 7, 614, 0)	(5, 1, 7, 614, 6, 709, 0)	(6, 1, 6, 709, 5, 787, 0)	(7, 1, 4, 841, 4, 913, 0)	(7, 2, 4, 913, 985, 0)	(8, 1, 4, 913, 985, 0)	(8, 2, 4, 985, 0)	(9, 1, 4, 7, 1057, 0)	(10, 1, 7, 1152, 0)	(10, 2, 7, 1247, 0)	(11, 1, 7, 1247, 0)	(11, 2, 7, 1247, 0)	
3	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 2, 213, 0)	(3, 1, 2, 213, 2, 307, 0)	(3, 2, 1, 401, 5, 520, 0)	(4, 1, 5, 520, 5, 574, 0)	(5, 1, 5, 574, 5, 628, 0)	(5, 2, 5, 628, 5, 682, 0)	(6, 1, 5, 682, 5, 736, 0)	(6, 2, 5, 736, 5, 790, 0)	(7, 1, 7, 844, 0)	(7, 2, 6, 844, 0)	(8, 1, 6, 994, 0)	(8, 2, 6, 994, 0)	(9, 1, 6, 7, 1072, 0)	(9, 2, 6, 7, 1167, 0)	(10, 1, 7, 1262, 0)	(11, 1, 7, 1262, 0)
4	(1, 1, 2, 0, 0)	(2, 1, 2, 94, 2, 188, 0)	(2, 2, 2, 188, 2, 282, 0)	(3, 1, 2, 282, 2, 376, 0)	(4, 1, 3, 470, 5, 564, 0)	(5, 1, 5, 564, 1, 618, 0)	(5, 2, 5, 618, 5, 737, 0)	(6, 1, 5, 737, 5, 791, 0)	(6, 2, 5, 791, 4, 845, 0)	(7, 1, 4, 845, 917, 0)	(7, 2, 4, 917, 0)	(8, 1, 4, 917, 0)	(8, 2, 4, 972, 0)	(9, 1, 7, 972, 0)	(9, 2, 7, 1090, 0)	(10, 1, 7, 1185, 0)	(11, 1, 7, 1185, 0)
5	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 2, 188, 0)	(2, 2, 2, 188, 2, 282, 0)	(3, 1, 2, 282, 2, 376, 0)	(4, 1, 3, 470, 5, 564, 0)	(5, 1, 5, 564, 4, 618, 0)	(5, 2, 5, 618, 5, 690, 0)	(6, 1, 5, 690, 6, 744, 0)	(6, 2, 6, 744, 6, 822, 0)	(7, 1, 6, 822, 900, 0)	(7, 2, 6, 995, 0)	(8, 1, 6, 995, 0)	(8, 2, 6, 995, 0)	(9, 1, 7, 7, 1067, 0)	(9, 2, 7, 7, 1162, 0)	(10, 1, 7, 1257, 0)	(11, 1, 7, 1257, 0)
6	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 3, 213, 0)	(3, 1, 2, 213, 2, 307, 0)	(3, 2, 2, 401, 1, 495, 0)	(4, 1, 1, 495, 3, 614, 0)	(5, 1, 3, 614, 4, 708, 0)	(6, 1, 4, 708, 6, 780, 0)	(6, 2, 6, 780, 6, 858, 0)	(7, 1, 6, 858, 7, 936, 0)	(7, 2, 6, 936, 1031, 0)	(8, 1, 6, 936, 0)	(8, 2, 6, 7, 1109, 0)	(9, 1, 6, 7, 1204, 0)	(10, 1, 7, 1204, 0)	(11, 1, 7, 1204, 0)	(11, 2, 7, 1204, 0)	
7	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 1, 213, 0)	(3, 1, 1, 213, 1, 332, 0)	(4, 1, 3, 332, 5, 451, 0)	(4, 2, 3, 451, 3, 505, 0)	(5, 1, 3, 505, 4, 599, 0)	(6, 1, 4, 599, 5, 671, 0)	(6, 2, 4, 671, 1, 725, 0)	(7, 1, 4, 725, 4, 844, 0)	(7, 2, 4, 844, 6, 916, 0)	(8, 1, 4, 916, 994, 0)	(8, 2, 4, 994, 0)	(9, 1, 7, 7, 1089, 0)	(9, 2, 7, 7, 1184, 0)	(10, 1, 7, 1184, 0)	(11, 1, 7, 1184, 0)	
8	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 1, 213, 0)	(3, 1, 2, 213, 2, 307, 0)	(3, 2, 2, 307, 5, 426, 0)	(4, 1, 5, 426, 1, 480, 0)	(4, 2, 5, 480, 4, 599, 0)	(5, 1, 4, 599, 4, 725, 0)	(5, 2, 4, 725, 5, 797, 0)	(6, 1, 4, 797, 4, 851, 0)	(6, 2, 4, 851, 923, 0)	(7, 1, 4, 923, 0)	(7, 2, 4, 995, 0)	(8, 1, 4, 995, 0)	(8, 2, 4, 995, 0)	(9, 1, 7, 995, 0)	(9, 2, 7, 995, 0)	
9	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 1, 213, 0)	(3, 1, 1, 213, 1, 332, 0)	(4, 1, 3, 332, 5, 451, 0)	(4, 2, 3, 451, 4, 505, 0)	(5, 1, 3, 505, 4, 577, 0)	(6, 1, 4, 577, 5, 649, 0)	(6, 2, 4, 649, 7, 703, 0)	(7, 1, 4, 703, 6, 798, 0)	(7, 2, 4, 798, 1, 876, 0)	(8, 1, 4, 876, 995, 0)	(8, 2, 4, 995, 0)	(9, 1, 4, 995, 0)	(9, 2, 4, 995, 0)	(10, 1, 7, 995, 0)	(10, 2, 7, 995, 0)	
10	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 3, 213, 0)	(3, 1, 3, 213, 3, 307, 0)	(3, 2, 2, 401, 3, 495, 0)	(4, 1, 3, 495, 1, 589, 0)	(4, 2, 3, 589, 4, 708, 0)	(5, 1, 4, 708, 6, 780, 0)	(5, 2, 4, 780, 7, 858, 0)	(6, 1, 4, 858, 4, 953, 0)	(6, 2, 4, 953, 2, 1025, 0)	(7, 1, 4, 1025, 0)	(7, 2, 4, 1025, 0)	(8, 1, 4, 1025, 0)	(8, 2, 4, 1025, 0)	(9, 1, 7, 1025, 0)	(9, 2, 7, 1025, 0)	
11	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 1, 213, 0)	(3, 1, 2, 213, 2, 332, 0)	(4, 1, 3, 332, 5, 426, 0)	(4, 2, 3, 426, 1, 480, 0)	(5, 1, 3, 480, 4, 599, 0)	(5, 2, 3, 599, 5, 671, 0)	(6, 1, 4, 671, 1, 725, 0)	(6, 2, 4, 725, 7, 820, 0)	(7, 1, 4, 820, 6, 915, 0)	(7, 2, 4, 915, 993, 0)	(8, 1, 4, 993, 0)	(8, 2, 4, 993, 0)	(9, 1, 4, 993, 0)	(9, 2, 4, 993, 0)	(10, 1, 7, 993, 0)	(10, 2, 7, 993, 0)
12	(1, 1, 2, 0, 0)	(2, 1, 1, 94, 1, 213, 0)	(3, 1, 3, 213, 3, 332, 0)	(4, 1, 3, 332, 5, 426, 0)	(4, 2, 3, 426, 1, 480, 0)	(5, 1, 3, 480, 7, 599, 0)	(5, 2, 3, 599, 5, 694, 0)	(6, 1, 4, 694, 7, 748, 0)	(6, 2, 4, 748, 4, 843, 0)	(7, 1, 4, 843, 4, 915, 0)	(7, 2, 4, 915, 987, 0)	(8, 1, 4, 987, 0)	(8, 2, 4, 987, 0)	(9, 1, 4, 987, 0)	(9, 2, 4, 987, 0)	(10, 1, 7, 987, 0)	(10, 2, 7, 987, 0)

表 4 预约机制下最优集卡调度方案作业信息表

Tab. 4 Job information form of optimal scheduling plan in appointment system

集卡编号	总行驶里程/ km	堆场总装箱时间/ min	堆场总等待时间/ min	码头总等待时间/ min	码头总操作时间/ min	CO ₂ 排放量/ kg	最后返回堆场 时间/min
1	1 080	48	0	89	91	2 887.20	1 308
2	1 120	45	0	83	94	2 992.53	1 342
3	1 040	48	0	93	104	2 783.07	1 357
4	1 060	45	0	83	92	2 833.31	1 280
5	1 140	48	0	91	98	3 047.19	1 352
6	1 080	42	0	82	95	2 886.20	1 299
7	1 060	42	9	84	93	2 833.20	1 288
8	940	39	0	69	82	2 512.00	1 090
9	880	36	0	68	83	2 352.66	1 067
10	940	36	0	68	75	2 510.78	1 119
11	900	36	0	67	86	2 405.89	1 065
12	880	36	0	70	73	2 351.78	1 059
13	1 000	39	0	70	84	2 671.33	1 091
14	800	36	0	72	69	2 139.90	1 062
15	860	30	67	61	71	2 296.90	1 089
16	800	30	67	59	67	2 137.24	1 048
17	840	36	67	76	68	2 245.89	1 087
18	780	33	67	68	69	2 085.79	1 017
19	900	33	67	65	77	2 404.34	1 119
20	720	27	197	50	69	1 924.13	1 063
21	760	30	187	56	70	2 031.24	1 103
22	740	27	197	51	66	1 976.91	1 081
23	740	27	197	53	58	1 976.25	1 075
24	760	33	197	59	65	2 031.35	1 114
25	560	21	317	47	41	1 496.05	986
26	580	21	307	46	46	1 549.49	1 000
27	560	21	307	43	42	1 495.71	973
28	580	21	307	40	54	1 549.71	1 002
29	560	21	307	43	41	1 495.60	972
30	560	21	317	44	46	1 496.27	988
31	580	24	307	50	49	1 550.59	1 010
32	520	18	391	34	46	1 388.83	1 009
33	480	18	437	42	28	1 281.72	1 005
34	200	9	623	20	15	534.86	867
合计	27 000	1 077	4 939	2 096	2 307	72 155.91	37 387

3.3 算例对比分析

在通过改进蚁群算法对预约机制下考虑碳排放的多码头集卡调度优化方案进行求解基础上,表 5、6 中给出了以往传统模式无预约机制下堆场面对同样的多集装箱码头集港任务时的集卡调度方案及相关的运输距离、作业时间、碳排放等数据. 同样的多码头集港需求,传统模式下调度方案需要堆场保有 39 辆集卡,其集卡最多集港次数为 11 次,单车最长行驶里程为 820 km,其平均堆场等待时间为 0 min,平均码头等待时间为 403.33 min,其最终返回堆场的平均时间为 1 182.41 min,集卡平均 CO₂ 排放量为 1 888.77 kg. 由表 4 预约机制最优集卡调度优化方案数据可知,其需求集卡保有量仅为 34 辆,调度方案中集卡最多服务次数为 16 次,单车最长行驶里程

为 1 140 km,堆场平均等待时间为 145.26 min,码头平均等待时间为 61.65 min,集卡最终返回堆场的平均时间为 1 099.62 min,集卡平均 CO₂ 排放量为 2 122.23 kg.

从调度方案中集卡数量来看,考虑预约机制的新调度模式仅为 34 台,较传统调度方案少 5 台;传统方案下 CO₂ 排放达 73 662.19 kg,基于预约机制优化方案 CO₂ 总排放量为 72 155.91 kg,碳排放量减少了 2.04%;预约机制下集卡总工作时间为 37 387 min,较传统模式减少 18.92%(传统模式下集卡工作总时间为 46 114 min);从等待时间方面来看,预约机制下集卡优化调度与传统模式最大的区别在于,预约机制下某集卡服务路径中等待时间多是在堆场产生的,而传统模式中集卡由于没有预约信息的沟通,只能在完成

上一次集港任务返回堆场后马上出发,前往下一码头进行集港,但往往其到达码头后由于码头方面种种原因需要在码头等待较长一段时间,急速工况下增加了燃油消耗以及 CO₂ 排放量。

由此可见,在基于预约机制的单堆场多码头集卡优化调度模式中,由于有对码头集港时间窗及各预约时段最大预约份额等预约信息的沟通,堆场在

安排集卡集港服务路径时可以更加方便地安排集卡的服务次序及作业流程,合理规避码头集港作业的高峰时段,避免码头闸口或港内堆场的过长等待.与传统集卡调度模式相比,本文所构建的调度优化方案能降低堆场对集卡数量的要求,提高集卡集港作业效率,有效减少能源消耗,降低集港作业过程中的 CO₂ 排放总量,最终减少堆场的运营成本。

表 5 传统模式送箱集卡调度方案

Tab. 5 Traditional truck scheduling plan

集卡 编号	送箱服务次序											
	1	2	3	4	5	6	7	8	9	10	11	12
1	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
2	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
3	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
4	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
5	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
6	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
7	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
8	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(6, 1, 4, 606,0)	(7, 1, 4, 678,0)	(8, 1, 1, 750,0)	(10,1,1, 869,0)	
9	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 3, 418,0)	(5, 1, 3, 512,0)	(7, 1, 4, 606,0)	(8, 1, 1, 759,0)	(10,1,1, 906,0)		
10	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 6, 418,0)	(5, 1, 3, 496,0)	(6, 1, 7, 590,0)	(7, 1, 1, 685,0)	(9, 1, 4, 804,0)	(11,1,1, 999,0)	
11	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 6, 418,0)	(5, 1, 3, 496,0)	(6, 1, 7, 590,0)	(7, 1, 1, 685,0)	(9, 1, 4, 804,0)	(11,1,1, 999,0)	
12	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 6, 418,0)	(5, 1, 3, 496,0)	(6, 1, 7, 590,0)	(7, 1, 1, 685,0)	(9, 1, 4, 804,0)	(11,1,1, 999,0)	
13	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 6, 418,0)	(5, 1, 3, 496,0)	(6, 1, 7, 590,0)	(7, 1, 1, 685,0)	(9, 1, 4, 804,0)	(11,1,1, 999,0)	
14	(1, 1, 5, 0,0)	(1, 2, 6, 54,0)	(2, 1, 7, 132,0)	(3, 1, 2, 227,0)	(4, 1, 2, 324,0)	(4, 2, 6, 418,0)	(5, 1, 3, 496,0)	(6, 1, 7, 590,0)	(7, 1, 1, 685,0)	(9, 1, 4, 804,0)	(11,1,1, 999,0)	
15	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(6, 2, 3, 662,0)	(7, 1, 3, 756,0)	(8, 1, 4, 850,0)	(8, 2, 4, 922,0)	(11,1,1, 994,0)	
16	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(6, 2, 3, 662,0)	(7, 1, 3, 756,0)	(8, 1, 4, 850,0)	(8, 2, 4, 922,0)	(11,1,1, 994,0)	
17	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(6, 2, 3, 662,0)	(7, 1, 3, 756,0)	(8, 1, 4, 850,0)	(8, 2, 4, 922,0)	(11,1,1, 994,0)	
18	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(6, 2, 3, 662,0)	(7, 1, 3, 756,0)	(8, 1, 4, 850,0)	(8, 2, 4, 922,0)	(11,1,1, 994,0)	
19	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(6, 2, 3, 662,0)	(7, 1, 3, 756,0)	(8, 1, 4, 850,0)	(8, 2, 4, 922,0)	(11,1,1, 994,0)	
20	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(6, 2, 3, 662,0)	(7, 1, 3, 756,0)	(8, 1, 4, 850,0)	(8, 2, 4, 922,0)	(11,1,1, 994,0)	
21	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(7, 1, 3, 662,0)	(7, 2, 3, 771,0)	(8, 1, 4, 865,0)	(10,1,1, 937,0)	(11,1,1, 1 146,0)	
22	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(5, 1, 4, 496,0)	(6, 1, 2, 568,0)	(7, 1, 3, 662,0)	(7, 2, 3, 771,0)	(8, 1, 4, 865,0)	(10,1,1, 937,0)	(11,1,1, 1 146,0)	

表 5(续)

集卡编号	送箱服务次序											
	1	2	3	4	5	6	7	8	9	10	11	12
23	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(4, 1, 4, 424,0)	(6, 1, 4, 496,0)	(6, 2, 2, 639,0)	(7, 1, 1, 733,0)	(9, 1, 1, 852,0)	(12,1,1, 1 026,0)			
24	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(5, 1, 4, 424,0)	(5, 2, 7, 519,0)	(6, 1, 3, 614,0)	(7, 1, 3, 708,0)	(8, 1, 7, 802,0)				
25	(1, 1, 5, 0,0)	(2, 1, 6, 151,0)	(3, 1, 7, 229,0)	(5, 1, 4, 424,0)	(5, 2, 7, 519,0)	(6, 1, 3, 614,0)	(7, 1, 3, 708,0)	(8, 1, 7, 802,0)				
26	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(6, 1, 7, 625,0)	(7, 1, 1, 720,0)	(9, 1, 1, 839,0)	(12,1,1, 1 026,0)				
27	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(6, 1, 7, 625,0)	(7, 1, 1, 720,0)	(9, 1, 1, 839,0)	(12,1,1, 1 026,0)				
28	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(6, 1, 7, 625,0)	(7, 1, 1, 720,0)	(9, 1, 1, 839,0)	(12,1,1, 1 026,0)				
29	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(6, 1, 7, 625,0)	(7, 1, 1, 720,0)	(9, 1, 1, 839,0)	(12,1,1, 1 026,0)				
30	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(6, 1, 7, 625,0)	(7, 1, 1, 720,0)	(9, 1, 1, 839,0)	(12,1,1, 1 026,0)				
31	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
32	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
33	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
34	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
35	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
36	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
37	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
38	(1, 1, 5, 0,0)	(4, 1, 7, 271,0)	(4, 2, 1, 412,0)	(5, 1, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)						
39	(1, 1, 5, 0,0)	(5, 1, 3, 391,0)	(5, 2, 2, 531,0)	(7, 1, 7, 625,0)	(9, 1, 1, 772,0)							

表 6 传统模式下集卡调度方案作业信息表

Tab. 6 Traditional job information form of scheduling plan

集卡编号	总行驶里程/ km	堆场总装箱时间/ min	堆场总等待时间/ min	码头总等待时间/ min	码头总操作时间/ min	CO ₂ 排放量/ kg	最后返回堆场 时间/min
1	820	33	0	224	69	2 209.02	1 146
2	820	33	0	224	69	2 209.02	1 146
3	820	33	0	224	69	2 209.02	1 146
4	820	33	0	224	69	2 209.02	1 146
5	820	33	0	224	69	2 209.02	1 146
6	820	33	0	224	69	2 209.02	1 146
7	820	33	0	224	69	2 209.02	1 146
8	820	33	0	224	69	2 209.02	1 146
9	760	30	0	291	65	2 056.65	1 146
10	820	33	0	338	75	2 222.29	1 266
11	820	33	0	338	75	2 222.29	1 266
12	820	33	0	338	75	2 222.29	1 266
13	820	33	0	338	75	2 222.29	1 266
14	820	33	0	338	75	2 222.29	1 266
15	760	33	0	413	60	2 069.92	1 266
16	760	33	0	413	60	2 069.92	1 266

表 6(续)

集卡编号	总行驶里程/ km	堆场总装箱时间/ min	堆场总等待时间/ min	码头总等待时间/ min	码头总操作时间/ min	CO ₂ 排放量/ kg	最后返回堆场 时间/min
17	760	33	0	413	60	2 069.92	1 266
18	760	33	0	413	60	2 069.92	1 266
19	760	33	0	413	60	2 069.92	1 266
20	760	33	0	413	60	2 069.92	1 266
21	800	33	0	366	66	2 171.38	1 265
22	800	33	0	366	66	2 171.38	1 265
23	680	27	0	615	64	1 880.02	1 386
24	560	24	0	272	41	1 521.24	897
25	560	24	0	272	41	1 521.24	897
26	680	24	0	620	62	1 880.02	1 386
27	680	24	0	620	62	1 880.02	1 386
28	680	24	0	620	62	1 880.02	1 386
29	680	24	0	620	62	1 880.02	1 386
30	680	24	0	620	62	1 880.02	1 386
31	480	18	0	486	42	1 332.34	1 026
32	480	18	0	486	42	1 332.34	1 026
33	480	18	0	486	42	1 332.34	1 026
34	480	18	0	486	42	1 332.34	1 026
35	480	18	0	486	42	1 332.34	1 026
36	480	18	0	486	42	1 332.34	1 026
37	480	18	0	486	42	1 332.34	1 026
38	480	18	0	486	42	1 332.34	1 026
39	380	15	0	600	31	1 078.39	1 026
合计	27 000	1 077	0	15 730	2 307	73 662.19	46 114

4 结论

在考虑集卡碳排放和码头预约机制的基础上合理调度集卡车队,完成各码头最终确定的各预约时段及其预约份额的集港任务,可以有效降低堆场集卡的调用数量,提高其运营效率,减少燃油消耗和 CO₂ 排放,降低堆场运营成本. 本文针对预约机制下单一堆场多集装箱码头间送箱集卡优化调度问题,在考虑碳排放基础上构建了相应的优化模型,设计了改进的蚁群算法对其进行求解. 算例求解及对比分析表明了本文所构建模型及算法的有效性. 研究成果拓展了堆场、码头相关的理论研究,为堆场企业的集卡调度优化决策和低碳港口建设的相关决策提供了一定参考.

应该指出,由于船舶集港所涉及各个环节和影响因素较为复杂,在求解分析过程中本文没有考虑集港过程中的集卡在港的排队规律、相关干扰因素以及集卡失约等问题,这些都有待于未来继续研究.

参考文献:

[1] 曾庆成, 张笑菊, 陈文浩, 等. 基于 BCMP 排队网络的码头集

卡预约优化模型[J]. 系统工程学报, 2013, 28(5): 592.

ZENG Qingcheng, ZHANG Xiaoju, CHEN Wenhao, *et al.* Optimization model for truck appointment based on BCMP queuing network[J]. Journal of Systems Engineering, 2013, 28(5): 592.

[2] NAMBOOTHIRIA R, ERERA A L. Planning local container drayage operations given a port access appointment system[J]. Transportation Research Part E: Logistics & Transportation Review, 2008, 44(2): 185.

[3] 蒋美仙, 张晓, 冯定忠, 等. 基于 Poisson 过程的港口码头集卡预约调度优化[J]. 浙江工业大学学报, 2016, 44(3): 292. JIANG Meixian, ZHANG Xiao, FENG Dingzhong, *et al.* Study on appointment scheduling model of container port truck based on poisson process [J]. Journal of Zhejiang University of Technology, 2016, 44(3): 292.

[4] HUYNH N, WATLON C M. Robust scheduling of truck arrivals at information on container terminals [J]. Journal of Transportation Engineering, 2008, 134(8): 347.

[5] 邵乾虔, 靳志宏, 邢磊. 基于联动预约模式的集装箱码头集疏港仿真优化[J]. 交通运输系统工程与信息, 2016, 16(4): 217.

SHAO Qianqian, JIN Zhihong, XING Lei. Simulation optimization of consolidation and configuration in container terminals based on cooperative appointment [J]. Journal of Transportation Systems Engineering and Information Technology, 2016, 16(4): 217.

[6] 许巧莉, 孙丽君, 胡祥培, 等. 非平稳到达的码头集卡预约优化模型[J]. 大连理工大学学报, 2014, 54(5): 589.

(下转第 1280 页)