

货滚码头甩挂运输系统内牵引车调度优化

杨珍花^{1,2}, 邢磊¹, 徐奇¹, 靳志宏^{1*}

(1. 大连海事大学交通运输工程学院, 辽宁 大连 116026; 2. 山东科技大学经济管理学院, 山东 青岛 266590)

摘要: 作业舱容限制与多船同步作业是导致货滚甩挂码头牵引车调度困难的关键原因. 为提高牵引车的运作效率, 降低系统运营成本, 建立了货滚甩挂码头牵引车调度的混合整数规划模型, 推导了目标函数的下界值. 同时, 开发模拟退火算法, 求解滚装船在港时间无交叉和有交叉两大类算例. 结果显示, 模拟退火算法所得方案可大幅度降低现有调度规则下的运营成本, 同时与下界值的差距较小, 证明了模型和算法的有效性. 而且, 当滚装船在港时间无交叉时算法的优化效果更好.

关键词: 甩挂运输; 货滚甩挂码头; 调度优化; 下界值; 模拟退火算法

中图分类号: U691+.31 文献标识码: A 文章编号: 1000-5781(2019)02-0252-14

doi: 10.13383/j.cnki.jse.2019.02.009

Scheduling optimization on tractor-and-trailer transportation system inside the ro-ro terminal

Yang Zhenhua^{1,2}, Xing Lei¹, Xu Qi¹, Jin Zhihong^{1*}

(1. College of Transportation Engineering, Dalian Maritime University, Dalian 116026, China;

2. College of Economics and Management, Shandong University of Science and Technology, Qingdao 266590, China)

Abstract: The constraint of cabin capacity and the synchronized operation of multi ships are key factors to complicated scheduling of tractors at the ro-ro tractor-and-trailer terminal. To improve the operation efficiency of tractors and reduce the operating cost of the ro-ro terminal as a whole, this paper establishes a mixed integer programming model for the tractor scheduling. A lower bound of the objective function is derived. By adopting the simulated annealing algorithm(SA), numerical solutions are obtained under two conditions: overlapping ship turnaround time and non-overlapping ship turnaround time. Results show that the operating cost obtained by SA is much lower than the cost of the practical operation rule. Also, the gap between the numerical results and the lower bound is small, which further proves the effectiveness of the model and the algorithm. Moreover, the algorithm performs better when the ships' turnaround time has no overlap.

Key words: tractor-and-trailer transportation; ro-ro tractor-and-trailer terminal; scheduling optimization; lower bound; simulated annealing algorithm

1 引言

甩挂运输中, 牵引车(带动力, 不可载货)和挂车(无动力, 可载货)可灵活耦合或分离. 当一段运输过程结

收稿日期: 2016-01-03; 修订日期: 2016-11-17.

基金项目: 国家自然科学基金资助项目(71572023; 71431001; 71302044; 71602130); 欧盟 FP7 居里夫人资助项目(ENRICH-612546); 中央高校基本科研业务费专项资金资助项目(3132016301; 3132018113).

*通讯作者

束时,二者从耦合状态变为分离状态,挂车停留,牵引车则从等待装卸中解放出来,继续执行其他任务或返回车场。这样,可以极大提高牵引车的利用率,降低其保有量。货滚运输则是使用半挂车、集装箱或汽车等承载货物,将运输工具和货物一起开上滚装船,到达目的地之后再从滚装船上开下的运输方式^[1],多用于海湾和海峡等短距离海运中。

为加强东北地区与山东省、华东地区的经济交流,2005年开始,交通部在渤海湾推行货滚甩挂运输。此举结合了两类运输方式的优势,既能打破公路运输绕行渤海湾的瓶颈,提高能效和人效,降低碳排放,又能提高货滚运输的安全性,加快航线两端腹地的货物集疏运。甩挂运输与货滚运输结合,形成了一种新的多子系统构成的货物运输复杂系统,而货滚甩挂码头是其中一个重要的组分,也是两种运输方式的衔接点。货滚甩挂码头涉及的资源种类多,数量大,同时又受到时间、空间的限制,其作业调度难度较两种方式单独运营都要大得多。但理论研究却相对滞后,仍集中在对甩挂运输这单一方式的作业调度上。

在我国,甩挂运输模式多为“1台牵引车+1台半挂车”,与此非常相似的是在欧洲与澳大利亚推广的“1台汽车(带动力,可载货)+1台挂车”的模式,针对该模式的研究称为TTRP问题。国外学者Chao^[2]首先提出TTRP问题,汽车拖带挂车形成汽车列车进行配送,且汽车停在客户点处等待装卸货。随后学者们开发不同的算法求解此问题,如Lin等^[3-5]运用模拟退火算法求解了带时间窗的TTRP问题;Villegas等^[6,7]提出混合元启发式方法进行求解;Derigs等^[8]研究了TTRP中汽车和挂车不可交换货物的问题,参考多人关于局部搜索(LS)和大邻域搜索(LNS)的方法,提出了一种较为复杂的启发式方法进行求解;胡志华等^[9]运用混合进化算法求解TTRP问题,并用2-opt对线路进行了优化。但TTRP问题涉及的车辆形式在我国鲜少使用。

针对我国常见模式的研究,Tan等^[10]以牵引车数量最小和路线费用最低为双重目标,构建了考虑外包任务的甩挂运输模型,并求解了问题的帕累托前沿。Lee等^[11]针对有时间窗约束的港口甩挂运输问题,建立了有外包任务的模型,得出外包任务提前选择的规律。Francis等^[12]提出变半径方法(VR)和贪婪随机过程(GRP)来求解有确定任务和灵活任务的多资源(牵引车和挂车)路径问题,但并未给出不确定任务的现实意义。Xue等^[13]考虑客户点处的拆装箱作业时间,将每个任务分成两个子任务,可由两台不同的牵引车执行,同时将所有节点拆分成两个虚拟节点。后续的研究中,Xue等^[14]设计了蚁群优化算法,Xue等^[15]开发了CBC算法解决大规模区域港口集装箱拖挂问题。胡志华等^[16]运用两阶段方法解决了集装箱空重箱的循环甩挂问题,后续胡志华^[17]又解决了集装箱码头间互拖的甩挂运输调度问题。钟慧玲等^[18]采用遗传算法及仿真手段解决了内河港口甩挂运输的选址问题。余莉等^[19]将甩挂运输网络转化为有向图并设计启发式算法进行求解。另外,文献[20]的问题本质与甩挂运输类似,其与本文的不同之处在于时间窗的设置。

至于两种运输方式相结合的货滚甩挂运输方面,国内均为针对鲁辽甩挂大通道或者烟大航线开展的前提条件或开通后的经济社会效益的论述、风险分析和防范措施,尚无调度方面的定量研究,国外也尚未找到相关文献。同样地,针对货滚甩挂码头的牵引车作业调度优化方面的研究也尚未出现。但在货滚码头的牵引车调度系统中,船舶在港作业的时间短,船舱内作业空间有限,若调度不利,极易造成船期延误,码头拥堵,资源浪费,货物集疏运延缓,进而增加货运公司成本,损害码头信誉。调度人员必须在有限的时间和设备资源允许的范围内,协调好牵引车在船舶上的挂车装卸、码头堆场和仓库内挂车提放等环节运行方案之间的配合,并力求降低整个系统的运作成本,提高运作效率。因此,针对这一方面的研究有着重要的现实意义。

烟大(烟台—大连)航线是渤海湾最为重要的货滚甩挂航线。本文依托大连湾货滚码头实际运作情况,考虑一个规划周期内多艘滚装船同时在港作业的现实因素,在船期、作业舱容两个维度的约束下,解决了挂车在货滚码头内部运输的任务指派和排序问题。首先,建立船舱作业空间有限制且带时间窗的货滚码头牵引车调度模型。其次,依据滚装船在港时间无交叉与有交叉两类问题各自的特点,分别推导了目标函数的下界值。同时,设计模拟退火算法进行求解,该算法通过现有调度规则生成初始解,采用双邻域搜索的方式进行迭代优化。最后,根据现实运营数据设计在港时间无交叉和有交叉的两大类算例,并分别获得现有规则的调度结果、下界值以及本文算法的求解结果,对比发现本文算法所得的调度方案可大幅度降低现有规则下的运营成本,且与下界值的差距较小,证明本文算法有较高的精度,在求解货滚码头牵引车调度优化问题方面

是有效的,且滚装船在港时间无交叉时,算法的效果更好。

本文所提出的问题结构与文献[10]相似,不同之处在于:1)从宏观背景来看,文献[10]是针对新加坡港集装箱的运营情况进行研究,而本文的研究对象则是大连湾货滚码头。虽然都采用甩挂模式,但前者车箱分离,后者车厢一体,这是决定两个问题不同的根本原因。2)文献[10]中集装箱和挂车都有20尺和40尺两种尺寸,存在挂车和集装箱的匹配问题,当前挂车与待运输集装箱尺寸不一致时需要访问挂车中心更换挂车;本文中挂车尺寸不对决策过程产生影响,更不需更换挂车。3)文献[10]每一项任务都分两个阶段执行,第一阶段将集装箱送到客户点,第二阶段将装/卸完的集装箱运走,且任务开始前及结束后挂车和集装箱都是分开存放的,需要在任务开始前取挂车、取集装箱,任务结束后再将挂车和集装箱分离,并把集装箱送还回场站;本文车厢一体,无论装货还是卸货,直接运输挂车即可,避免了大量无效操作。4)文献[10]的集装箱运输是在港区后方进行的,不涉及上下船作业;本文中大多数任务的端点就是滚装船,船舱空间的局限性就成为了一个重要的制约条件。5)文献[10]中运输能力是由内部牵引车的数量决定的,完成不了的任务可以外包,且挂车数量也是有限的;本文所有任务由港口内部牵引车完成,牵引车数量足够,由于挂车即为待运输任务,不存在这一资源限制。可以看出,本文与文献[10]虽在运作模式上相似,但在具体流程上仍存在较大的差异。当前除文献[10]外,并无其他研究提出类似问题。因此,本文在问题结构上有着较强的创新性。

2 问题描述

货滚码头甩挂运输系统主要由三部分组成:堆场、仓库和滚装船。其中堆场用于存放待装船和卸船后待疏运的挂车;仓库为需要拼货的挂车提供拼装服务。系统的整体布局及挂车上船作业流程见图1。

首先,外部牵引车将装好的挂车送至仓库或堆场,其中需要进行拆挂的送至仓库进行处理,其他的则直接送至堆场等待运输,牵引车离开。滚装船靠岸开始作业后,从外部调集牵引车进入系统,先将船上的挂车卸至仓库或堆场。下船作业结束并完成维护工作后,开始进行上船作业。

牵引车从堆场或仓库,拖带挂车至滚装船,需注意到,受滚装船舱作业空间的限制,同时在船舱内脱挂的牵引车数量有一定限制,超出限制数量到达的牵引车需在滚装船外等待,待舱内牵引车下船后方可进入。牵引车在舱内将挂车甩下后,返回堆场或仓库重新挂上新的挂车继续上船作业。同时,由于仓库面积和作业设施的限制,部分需要拆装和已经完成拆装的挂车会在堆场暂存。因此在堆场和仓库之间也存在挂车的运输。所有任务完成后,牵引车离开。

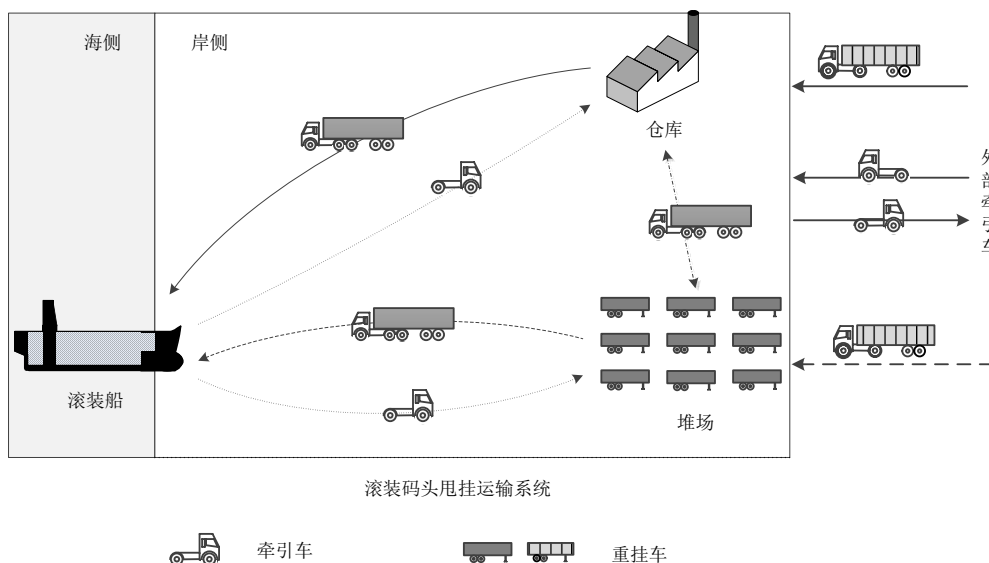


图1 货滚码头甩挂运输系统布局及上船作业流程

Fig. 1 Layout of the ro-ro tractor-and-trailer terminal and roll-on operation process

在一个规划期内,有多艘滚装船按照固定的班次到达和离开码头.每艘都有特定的上下船任务,同时停泊作业的时间也是有限的,牵引车需在相应的时间内完成其所有的上下船任务.这部分任务涉及滚装船与堆场、滚装船与仓库之间的挂车交流,且开始和结束时间均有着严格限制,称为滚装船直接相关任务(以下简称为直接任务).而仓库和堆场之间的挂车交流在整个规划期内均可,此部分任务称为滚装船间接相关任务(以下简称为间接任务).

同时,当滚装船的在港时间产生交叉时,会存在有几艘船同时作业的情况,所有的任务都需要利用有限的牵引车资源有序高效地完成.以大连至烟台滚装船发班时间为例,规划期内(第一天 8:00 至第二天 8:00)滚装船在港停留情况见下图 2 所示.

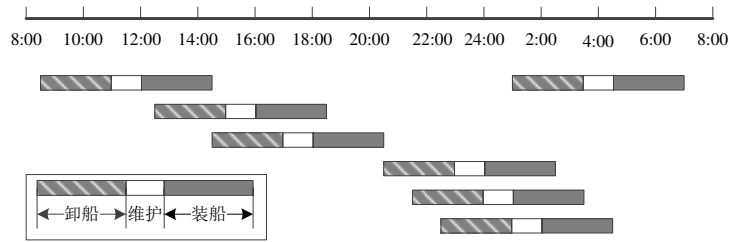


图 2 规划期内船舶在港停留时间分布

Fig. 2 Berth time distribution of ro-ro ships in the planning period

3 牵引车作业调度的确定性模型

3.1 假设条件

规划期内滚装船的班次数量以及每班次的靠港时间、离港时间、上下船作业时间是确定的;每个班次需要装卸的挂车数已知;靠港作业时间足以满足所有任务的操作需求;牵引车资源能够满足所有作业需求;一辆牵引车一次只能拖带一辆挂车;牵引车在滚装船之间、滚装船与堆场、滚装船与仓库、堆场与仓库之间的行驶时间是确定的.

3.2 参数和变量

本文设置 0 为规划期开始时间; T 为规划期长度; P 为规划期内滚装船班次总数; p 为第 p 班到达的滚装船编号; t_p 为第 p 班滚装船的靠港时间; T_1 为滚装船下船作业持续时间; T_2 为滚装船维护作业持续时间; T_3 为滚装船上船作业持续时间; $N_{p\alpha}^\beta$ 表示第 p 班滚装船的直接任务集合,可根据任务不同的流向(上船任务或下船任务)、不同的端点(堆场或仓库)划分为四个子集,分别由 α 和 β 的不同取值确定,其中 α 不同的取值区分不同的任务流向, β 不同的取值区分不同的任务端点.

$$N_{p\alpha}^\beta = \begin{cases} \text{第 } p \text{ 班滚装船的所有拖至堆场的下船任务集合,} & \alpha = 1, \beta = 1 \\ \text{第 } p \text{ 班滚装船的所有拖至仓库的下船任务集合,} & \alpha = 1, \beta = 2 \\ \text{第 } p \text{ 班滚装船的所有从堆场出发的上船任务集合,} & \alpha = 2, \beta = 1 \\ \text{第 } p \text{ 班滚装船的所有从仓库出发的上船任务集合,} & \alpha = 2, \beta = 2. \end{cases} \quad (1)$$

设 N_1 为规划期内从堆场运至仓库的任务集合; N_2 为规划期内从仓库运至堆场的任务集合; N 为规划期内所有任务集合, $N = N_1 \cup N_2 \cup (\bigcup_{p,\alpha,\beta} N_{p\alpha}^\beta)$; i 为任务编号;为统一形式,设置两项虚拟任务, U_1 表示牵引车刚开始工作的虚拟任务, U_2 表示牵引车结束工作的虚拟任务; D 为滚装船舱内部可同时容纳作业的牵引车数; T_Δ 为挂/摘一个挂车的时间; T_∇ 为牵引车在系统内任意两点之间移动所需的时间; T_{r-r} 为一台牵引车取(送)一个挂车在滚装船上消耗的时间; K 为牵引车总数; k 为牵引车编号; f 为牵引车的固定成本,包括车辆的折旧、驾驶人员的工资等; c_1 为牵引车运营单位时间的可变成本; c_2 为牵引车等待单位时间的机会

成本; $[E_i, L_i]$ 为任务 i 执行的时间窗, 其定义分别为

$$E_i = \begin{cases} t_p, & i \in N_{p1}^\beta, \beta = 1, 2 \\ t_p + T_1 + T_2, & i \in N_{p2}^\beta, \beta = 1, 2 \\ 0, & i \in N_1 \cup N_2, \end{cases} \quad (2)$$

$$L_i = \begin{cases} t_p + T_1, & i \in N_{p1}^\beta, \beta = 1, 2 \\ t_p + T_1 + T_2 + T_3, & i \in N_{p2}^\beta, \beta = 1, 2 \\ T, & i \in N_1 \cup N_2. \end{cases} \quad (3)$$

设置三个决策变量, t_k^0 为牵引车 k 到达系统的时间; $Y_{ik} \in \{0, 1\}, i \in N, \forall k$, 当任务 i 由牵引车 k 完成时, 其值为 1, 否则为 0; $Z_{ijk} \in \{0, 1\}, i, j \in N, \forall k$, 当牵引车 k 连续服务 i, j 两项任务时, 其值为 1, 否则为 0.

t_i 为执行任务 i 所需的时间. t_{ij} 为任务 i 和任务 j 之间的作业准备时间, 即牵引车从任务 i 终点行驶到任务 j 起点的时间跨度, 以一艘滚装船为例, 系统内共涉及 6 种任务类型, 任意两项任务进行组合, 共形成 36 种情况.

$$t_{ij} = \begin{cases} 0, & (i, j) \in \{(N_{p1}^1, N_1), (N_{p1}^2, N_2), (N_1, N_{p2}^2), (N_1, N_2), (N_2, N_{p2}^1), (N_2, N_1)\}, Z_{ijk} = 1, \forall p \\ \infty, & (i, j) \in \{(N_{p1}^\beta, N_{p2}^\beta), (N_{p2}^\beta, N_{p1}^\beta)\}, \beta = 1, 2, Z_{ijk} = 1, \forall p \\ T_\nabla, & \text{其它}, Z_{ijk} = 1, \forall p. \end{cases} \quad (4)$$

注意到, 针对单一滚装船, 不存在上下船作业同时进行的情况, 因此上下船任务之间的作业准备时间为 ∞ . 当存在两艘滚装船同时作业且二者作业时间有重合时, 若连续的两项上下船任务分别涉及到第 p 班滚装船和第 p^* 班滚装船且 $t_p \leq t_{p^*}, Z_{ijk} = 1, \beta = 1, 2, \forall (p, p^*)$, 作业准备时间变为

$$t_{ij} = \begin{cases} T_\nabla, & (i, j) \in \{(N_{p1}^\beta, N_{p^*1}^\beta), (N_{p^*1}^\beta, N_{p1}^\beta)\}, t_{p^*} < t_p + T_1 \\ T_\nabla, & (i, j) \in \{(N_{p2}^\beta, N_{p^*1}^\beta), (N_{p^*1}^\beta, N_{p2}^\beta)\}, t_{p^*} < t_p + T_1 + T_2 < t_{p^*} + T_1 \\ T_\nabla, & (i, j) \in \{(N_{p2}^\beta, N_{p^*2}^\beta), (N_{p^*2}^\beta, N_{p2}^\beta)\}, t_{p^*} + T_1 + T_2 < t_p + T_1 + T_2 + T_3 \\ \infty, & \text{其它}. \end{cases} \quad (5)$$

j_m^k 为牵引车 k 执行的第 m 项任务的编号,

$$j_m^k = \begin{cases} \sum_{j \in N} j Z_{U_1jk}, & m = 1, \forall k \\ \sum_{j \in N} j Z_{j_{m-1}^k j k}, & m = 2, 3, \dots, \sum_{i \in N} Y_{ik}, \forall k. \end{cases} \quad (6)$$

$t_{j_m^k}^0$ 为牵引车 k 开始执行第 m 项任务的时间,

$$t_{j_m^k}^0 = \begin{cases} t_k^0, & m = 1, j_m^k \in N, \forall k \\ t_{j_{m-1}^k}^0 + t_{j_{m-1}^k} + t_{j_{m-1}^k j_m^k}, & m = 2, 3, \dots, \sum_{i \in N} Y_{ik}, j_m^k \in N, \forall k. \end{cases} \quad (7)$$

$N_{p\alpha}^{\beta*} = w_p^1, w_p^2, \dots, w_p^u, \dots, w_p^{|N_{p\alpha}^\beta|}, t_{w_p^u}^0 \leq t_{w_p^u}^0$ 为第 p 班滚装船上(下)船任务的执行顺序. 其中 w_p^u 是第 u 项作业的任务序号.

t_i^0 为牵引车到达任务 i 起点的时间,

$$t_i^0 = \begin{cases} t_k^0, & Z_{U_1ik} = 1, k = 1, 2, 3, \dots, K \\ t_j^5 + t_{ij}, & Z_{jik} = 1, k = 1, 2, 3, \dots, K, j \in N. \end{cases} \quad (8)$$

t_i^1 为在任务 i 起点的等待时间, 仅在牵引车执行直接任务时才会产生. 当任务 i 为下船任务时, 导致牵引车产生等待的原因有两个, 一是牵引车到达时间早于该任务的时间窗, 需等待至时间窗较小时刻; 二是船舱内已经存在的牵引车数已达上限, 此时到达滚装船的牵引车需等待船舱内的牵引车下船. 因此, 针对下船任务 w_p^u , 牵引车在其起点的等待时间为

$$t_{w_p^u}^1 = \begin{cases} E_{w_p^u} - t_{w_p^u}^0, & t_{w_p^u}^0 < E_{w_p^u}, w_p^u \in N_{p1}^\beta, \beta = 1, 2, \forall p \\ 0, & u = 1, 2, \dots, D, t_{w_p^u}^0 \geq E_{w_p^u}, \forall p \\ \max\{0, t_{w_p^u}^{u-D} - t_{w_p^u}^0\}, & u = D+1, D+2, \dots, |N_{p1}^\beta|, \beta = 1, 2, t_{w_p^u}^0 \geq E_{w_p^u}, \forall p. \end{cases} \quad (9)$$

针对上船任务 i , 牵引车在其起点产生等待的原因则是由于牵引车到达的时间早于该任务的时间窗. 此时牵引车在起点的等待时间为

$$t_i^1 = \begin{cases} E_i - t_i^0, & t_i^0 < E_i, i \in N_{p2}^\beta, \beta = 1, 2, \forall p \\ 0, & t_i^0 \geq E_i, i \in N_{p2}^\beta, \beta = 1, 2, \forall p. \end{cases} \quad (10)$$

t_i^2 为牵引车离开任务 i 起点的时间,

$$t_i^2 = \begin{cases} t_i^0 + t_i^1 + T_{r-r}, & i \in N_{p1}^\beta, \beta = 1, 2, \forall p \\ t_i^0 + T_\Delta + t_i^1, & i \in N|N_{p1}^\beta|, \beta = 1, 2, \forall p. \end{cases} \quad (11)$$

t_i^3 为牵引车到达任务 i 终点的时间, $t_i^3 = t_i^2 + T_\nabla$.

t_i^4 为在任务 i 终点的等待时间, 由于在任务起点已经将执行时间约束在时间窗内, 因此不会出现早于时间窗到达而产生等待的情况. 只有执行上船任务, 船舱内已经存在的牵引车数已达上限时, 到达滚装船的牵引车才需等待, 即针对上船任务 w_p^u , 牵引车在其终点的等待时间为

$$t_{w_p^u}^4 = \begin{cases} 0, & u = 1, 2, \dots, D, \forall p \\ \max\{0, t_{w_p^u}^{5-u} - t_{w_p^u}^3\}, & u = 4, 5, \dots, |N_{p2}^\beta|, \beta = 1, 2, \forall p. \end{cases} \quad (12)$$

t_i^5 为牵引车离开任务 i 终点的时间, 如下

$$t_i^5 = \begin{cases} t_i^3 + t_i^4 + T_{r-r}, & i \in N_{p2}^\beta, \beta = 1, 2, \forall p \\ t_i^3 + t_i^4 + T_\Delta, & i \in N|N_{p1}^\beta|, \beta = 1, 2, \forall p. \end{cases} \quad (13)$$

t_k^{end} 为牵引车 k 结束工作的时间.

牵引车作业调度的确定性模型为

$$\text{Min} \sum_{k=1}^K \left[\frac{1}{|N|} \sum_{i \in N} Y_{ik} \right] f + c_1 \sum_{k=1}^K \sum_{i,j \in N} (t_i + t_{ij} Z_{ijk}) + c_2 \sum_{p=1}^P \sum_{\alpha, \beta=1,2} \sum_{i \in N_{p\alpha}^\beta} (t_i^1 + t_i^4), \quad (14)$$

s.t.

$$t_i^0 \geq t_k^0 Y_{ik}, \quad i \in N, \forall k, \quad (15)$$

$$t_i^5 \leq t_k^{\text{end}} Y_{ik}, \quad i \in N, \forall k, \quad (16)$$

$$0 \leq t_k^0, t_k^{\text{end}} \leq T, \quad \forall k, \quad (17)$$

$$(t_i^5 + t_{ij}) Z_{ijk} \leq t_j^0, \quad i, j \in N, \forall k, \quad (18)$$

$$\sum_{k=1}^K Y_{ik} = 1, \quad i \in N, \quad (19)$$

$$t_{w_p^u}^0 + t_{w_p^u}^1 \leq t_{w_p^{u+1}}^0 + t_{w_p^{u+1}}^1, \quad u = 1, 2, \dots, |N_{p1}^\beta|, \alpha, \beta = 1, 2, \forall p, \quad (20)$$

$$t_{w_p^u}^3 + t_{w_p^u}^4 \leq t_{w_p^{u+1}}^3 + t_{w_p^{u+1}}^4, \quad u = 1, 2, \dots, |N_{p2}^\beta|, \alpha, \beta = 1, 2, \forall p, \quad (21)$$

$$t_{w_p^{u+D}}^0 + t_{w_p^{u+D}}^1 \geq t_{w_p^u}^2, \quad u = 1, 2, \dots, |N_{p1}^\beta| - D, \beta = 1, 2, \forall p, \quad (22)$$

$$t_{w_p^{u+D}}^3 + t_{w_p^{u+D}}^4 \geq t_{w_p^u}^5, \quad u = 1, 2, \dots, |N_{p2}^\beta| - D, \beta = 1, 2, \forall p, \quad (23)$$

$$E_i \leq t_i^0 + t_i^1 \leq L_i, \quad i \in N, \quad (24)$$

$$E_i \leq t_i^5 \leq L_i, \quad i \in N, \quad (25)$$

$$\sum_{i \in N} Z_{ijk} \leq Y_{jk}, \quad j \in N, \forall k, \quad (26)$$

$$\sum_{j \in N} Z_{ijk} \leq Y_{ik}, \quad i \in N, \forall k, \quad (27)$$

$$Y_{ik}, Z_{ijk} \in \{0, 1\}, \quad i, j \in N, \forall k, \quad (28)$$

$$t_i = t_i^5 - t_i^0, \quad i \in N. \quad (29)$$

式(14)以整体调度的固定成本、可变成本以及机会成本之和最小为目标函数, 其中 $\left\lceil \frac{1}{|N|} \sum_{i \in N} Y_{ik} \right\rceil$ 的取值为 0 或 1, 仅当其值为 1 时表示牵引车 k 被使用, 此时牵引车 k 的固定成本才会包含在目标函数中; 式(15)和式(16)联合限制任务的执行时间不违反牵引车工作时间要求; 式(17) 保证牵引车的工作时间在规划期之内; 式(18)保证两项任务的执行时间不冲突; 式(19)限制每项任务由且仅由一台牵引车执行; 式(20)和式(21)保证先到达滚装船的牵引车优先上船; 式(22)和式(23)表示船上同时作业的挂车数量不能超过空间限制; 式(24)和式(25)保证任务的作业时间不违反时间窗约束; 式(26) 和式(27) 联合限制一项任务至多有一项紧前或紧后任务; 式(28) 是决策变量的取值约束; 式(29)是任务执行所需时间的推导。

4 下界值推导

目标函数是令牵引车使用的固定成本、可变成本与机会成本之和最小, 最理想的状态便是使用最少的牵引车并达到最大的利用率, 即避免发生牵引车等待, 降低相邻两项任务的作业准备时间. 令执行一项直接任务所需的时间为 $\Lambda = T_\Delta + 2T_\nabla + T_{r-r}$, 执行一项间接任务的时间为 $\Psi = 2T_\Delta + T_\nabla$. 下界值的推导主要分为滚装船在港时间无交叉和有交叉两种情况.

4.1 滚装船在港时间无交叉

牵引车数量首先应当保证能够完成所有任务; 其次由于直接任务仅能在要求的时间窗内执行的特殊性, 牵引车数量应同时保证能够完成直接任务; 第三, 当滚装船数量较多时, 留给间接任务的执行时间有限, 还要考虑到保证这部分任务完成的限制. 因此, 牵引车数量下界从这三方面进行求解.

1) 保证所有任务完成的下界

$$LB_{V1} = \left\lceil \frac{\Lambda}{T} \sum_{p=1}^P \sum_{\alpha, \beta=1,2} |N_{p\alpha}^\beta| + \Psi |N_1 \cup N_2| \right\rceil. \quad (30)$$

2) 保证直接任务完成的下界

$$LB_{V2} = \max \left\{ \left\lceil \max\{|N_{p1}^\beta|\} \Lambda / T_1 \right\rceil, \left\lceil \max\{|N_{p2}^\beta|\} \Lambda / T_3 \right\rceil \right\}, \quad \forall p, \beta = 1, 2. \quad (31)$$

3) 保证间接任务完成的下界

$$LB_{V3} = \left\lceil \Psi |N_1 \cup N_2| / (T_2 P + T_{\text{non}} + T_{\text{left}}) \right\rceil, \quad (32)$$

其中 T_{non} 为规划期内无滚装船在港停留的时间, T_{left} 是上下船作业期间完成直接任务后剩余的运力.

初始令 $T_{\text{left}} = 0$, 反复迭代求得其最终值

$$T_{\text{left}} = \sum_{p=1}^P \sum_{\beta=1,2} \left(\text{LB}_V(T_1 + T_2) - \Lambda(|N_{p1}^\beta| + |N_{p2}^\beta|) \right). \quad (33)$$

牵引车数量的下界 $\text{LB}_V = \max\{\text{LB}_{V1}, \text{LB}_{V2}, \text{LB}_{V3}\}$.

牵引车总运营时间为执行所有任务必需的时间与任务之间作业准备时间之和.

1) 任务本身必需的时间

$$\text{LB}_{T1} = \Lambda \sum_{p=1}^P \sum_{\alpha,\beta=1,2} |N_{p\alpha}^\beta| + \Psi |N_1 \cup N_2|. \quad (34)$$

2) 任务之间的作业准备时间主要是由间接任务两个方向任务数量的不平衡引起的. 上下船作业期间牵引车可以在一定程度上进行修补, 修补的数量为

$$N_f = \sum_{p=1}^P \left(\left\lfloor \frac{T_1}{\Psi} - \frac{|N_{p1}^\beta| \Lambda}{\Psi \text{LB}_V} \right\rfloor \right) \text{LB}_V + \sum_{p=1}^P \left(\left\lfloor \frac{T_3}{\Psi} - \frac{|N_{p2}^\beta| \Lambda}{\Psi \text{LB}_V} \right\rfloor \right) \text{LB}_V, \beta = 1, 2. \quad (35)$$

总体不平衡程度 $N_u = ||N_1| - |N_2||$, 则作业准备时间下界 $\text{LB}_{T2} = T_\nabla(N_u - N_f)$.

牵引车总的运营时间下界 $\text{LB}_T = \text{LB}_{T1} + \text{LB}_{T2}$.

机会成本是由于牵引车在滚装船处排队等待或早于任务时间窗到达等待造成的.

1) 每处理一项直接任务牵引车需要耗费的时间为 $\Lambda = T_{r-r} + T_\Delta + 2T_\nabla$. 船舱可同时容纳 D 台牵引车作业, 视为共有 D 条并列通道且相互之间无干扰. 以一艘滚装船的下船任务为例, 单一通道牵引车不发生等待时, 连续两台牵引车工作的时间关系如图3所示. η_k 和 η'_k 分别为第 k 台牵引车进和出船舱的时间.

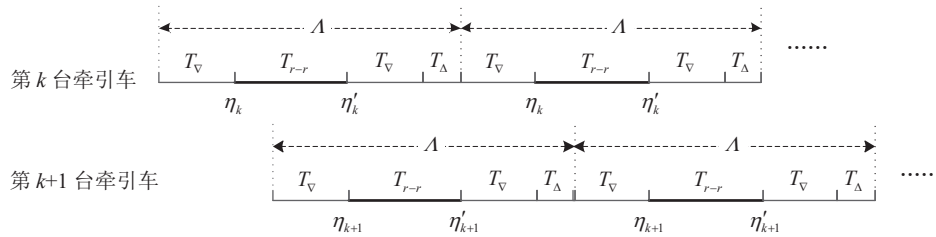


图3 同一通道的连续两台牵引车进出时间关系

Fig. 3 In-out time of two successive tractors in the same channel

可见, 单一通道在不发生等待的前提下可处理的任務数最多为 $\lfloor (2T_\nabla + T_\Delta)/T_{r-r} \rfloor + 1$, 由于共有 D 条并列通道, 因此单个滚装船期牵引车在不等待的前提下可处理的任務数最多为 $D(\lfloor (2T_\nabla + T_\Delta)/T_{r-r} \rfloor + 1)$. 同时也可以看出, 这是滚装船依资源限制所能容纳的最大的挂车数量. 所以在滚装船处, 牵引车在理想状态下是不会发生拥堵等待的.

2) 牵引车早于时间窗到达引起的等待时间, 主要是因为间接任务的数量不足以填补执行直接任务的空缺, 由直接任务的时间窗限制造成的, 其值 $\text{LB}_O = \max\{0, T_2P + T_{\text{non}} + T_{\text{left}} - \Psi|N_1 \cup N_2|\}$.

目标函数下界 $\text{LB} = f\text{LB}_V + c_1\text{LB}_T + c_2\text{LB}_O$.

4.2 滚装船在港时间有交叉

保证直接任务完成的下界

令一个批次直接任务(一艘滚装船的所有上/下船任务)的时间窗对应一个时段. 找出每两个时段交叉的时长, 从中寻找最小值, 表示为 ΔT , 令其作为单位时长将整个规划期离散化处理为 $S = \lceil T/\Delta T \rceil$ 个区段. 分配至第 s 区段内执行的下船任务数量 Num_s^1 和上船任务数量 Num_s^2 分别为

$$\text{Num}_s^1 = \sum_{p=1}^P \sum_{\beta=1,2} \left\lceil N_{p1}^\beta \frac{\Delta T}{T_1} \right\rceil, \quad (s-1)\Delta T \geq t_p \text{ 且 } s\Delta T \leq t_p + T_1,$$

$$\text{Num}_s^2 = \sum_{p=1}^P \sum_{\beta=1,2} \left\lceil N_{p2}^\beta \frac{\Delta T}{T_3} \right\rceil, \quad (s-1)\Delta T \geq t_p + T_1 + T_2 \text{ 且 } s\Delta T \leq t_p + T_1 + T_2 + T_3.$$

因此, 保证完成直接任务的牵引车数量下界为

$$\text{LB}_{V2}^* = \left\lceil \frac{\max_{s=1}^S \{\text{Num}_s^1 + \text{Num}_s^2\} \Delta T}{\Delta T} \right\rceil, \quad (38)$$

滚装船在港时间有交叉的牵引车数量下界 $\text{LB}_V^* = \max\{\text{LB}_{V1}, \text{LB}_{V2}^*, \text{LB}_{V3}\}$.

因此, 目标函数下界 $\text{LB}^* = f\text{LB}_V^* + c_1\text{LB}_T + c_2\text{LB}_O$.

5 算法设计

本文将同一艘滚装船的所有上(下)船任务定义为同一批次任务, 堆场与仓库之间的任务整体视为一个批次. 鉴于所有任务的时间窗较为宽泛, 算法的邻域搜索空间较大, 可操作性强, 同时模拟退火算法的求解精度高, 且内嵌的 Metropolis 准则可直接避免算法陷入局部最优解. 因此, 本文采用模拟退火算法进行求解.

5.1 编码方式

本文直接以任务为对象采用整数编码, 如图4所示. 牵引车所执行的挂车运输任务是相互独立且执行过程不可中断的, 每一位对应一项任务的编号, 每一行形成一台牵引车的任务执行序列, 整体则构成了规划期内的调度方案. 图4显示了4台牵引车执行25项任务的调度方案.

任务序列1	24	8	21	25	15	11	14	6	
任务序列2	2	23	22	16	19				
任务序列3	10	7	4	9	3	12	9	5	20
任务序列4	18	13	17	1					

图4 编码方式

Fig. 4 Encoding mode

5.2 生成初始解

所有直接任务根据批次划分为 $2P$ 个集合, 同时令 $N_1 \cup N_2$ 为第 $2P+1$ 个集合. r 为所有集合的编号, 第 r 个集合表示为 N_r .

为了确定每批直接任务执行的先后顺序并保证船舱内同时执行任务的牵引车数量不超过限制, 本文为调度方案设置附属矩阵 $\mathbf{M}$, $\mathbf{M}$ 共包含 $|N|$ 个模块, 每项任务有7个参数被记录在同一模块中, 分别为任务编号、执行该任务的牵引车编号、任务批次、任务开始执行时间、牵引车上滚装船时间、牵引车离开滚装船时间以及任务结束时间. 其中, 间接任务的第4项与第5项, 第6项与第7项分别相等.

按照如下规则生成初始解

步骤1 集合编号 $r \leftarrow 1$, 任务序列编号 $l \leftarrow 1$;

步骤2 判断 $r \leq 2P+1$ 是否成立, 若成立, 执行步骤3, 否则输出当前调度方案, 并进行附属矩阵的调整和任务序列的合并(具体见本节“对附属矩阵的处理”和“合并任务序列”);

步骤3 判断 N_r 是否为空, 若为空, $r \leftarrow r+1$, $l \leftarrow l+1$, 返回步骤2, 否则执行步骤4;

步骤4 随机从 N_r 中选择一项任务, 若当前序列尚无任务, 直接将其作为第一项任务置入, 该序列的开

始执行时间即为该项任务时间窗的较小值. 若序列中已有任务, 判断该任务置于当前序列末端是否会违反其时间窗约束, 若不违反, 将此任务置于当前序列末端; 若违反, $l \leftarrow l + 1$, 将此任务置入新的任务序列, 新序列的开始执行时间即为该项任务时间窗的较小值. 将该任务的参数记录到附属矩阵 M 的相应位置, 并将其从集合 N_r 中移出.

对附属矩阵的处理

针对每一批直接任务, 根据调度方案每项任务到达和离开滚装船的时间都已有相关记录, 基于到达滚装船的时间将附属矩阵 M 中的相关模块进行升序排列(若存在数值相同的任务, 对其进行随机排列), 则形成了任务的执行顺序. 此时, 若存在超出数量上限的牵引车同时在滚装船舱内作业的情况, 依照模型中式(22)和式(23)依次将超出限制的任务的执行时间后延, 相应地, 其它发生变化的任务相关参数也需要进行调整.

调整后若导致部分任务的执行时间晚于时间窗要求, 将相应任务依执行时间先后置于一新的任务序列中. 以一艘滚装船的下船任务为例, 初始解生成过程中附属矩阵的处理如下图5所示.

1,1,(8:00,8:15)	2,1,(8:30,8:45)	3,1,(9:00,9:15)	4,1,(9:30,9:45)	1,1,(8:00,8:15)	5,2,(8:00,8:15)	9,3,(8:00,8:15)	13,4,(8:00,8:15)
5,2,(8:00,8:15)	6,2,(8:30,8:45)	7,2,(9:00,9:15)	8,2,(9:30,9:45)	17,5,(8:00,8:15)	2,1,(8:30,8:45)	6,2,(8:30,8:45)	10,3,(8:30,8:45)
9,3,(8:00,8:15)	10,3,(8:30,8:45)	11,3,(9:00,9:15)	12,3,(9:30,9:45)	14,4,(8:30,8:45)	18,5,(8:30,8:45)	3,1,(9:00,9:15)	7,2,(9:00,9:15)
13,4,(8:00,8:15)	14,4,(8:30,8:45)	15,4,(9:00,9:15)	16,4,(9:30,9:45)	11,3,(9:00,9:15)	15,4,(9:00,9:15)	19,5,(9:00,9:15)	4,1,(9:30,9:45)
17,5,(8:00,8:15)	18,5,(8:30,8:45)	19,5,(9:00,9:15)	20,5,(9:30,9:45)	8,2,(9:30,9:45)	12,3,(9:30,9:45)	16,4,(9:30,9:45)	20,5,(9:30,9:45)
(步骤一)				(步骤二)			

1,1,(8:00,8:15)	5,2,(8:00,8:15)	9,3,(8:00,8:15)	13,4,(8:15,8:30)
17,5,(8:15,8:30)	2,1,(8:30,8:45)	6,2,(8:30,8:45)	10,3,(8:30,8:45)
14,4,(8:45,9:00)	18,5,(8:45,9:00)	3,1,(9:00,9:15)	7,2,(9:00,9:15)
11,3,(9:00,9:15)	15,4,(9:15,9:30)	19,5,(9:15,9:30)	4,1,(9:30,9:45)
8,2,(9:30,9:45)	12,3,(9:30,9:45)	16,4,(9:45,10:00)	20,5,(9:45,10:00)
(步骤三)			

图5 附属矩阵的处理

Fig. 5 Processing of the affiliated matrix

下船作业时间为 8:00~10:00, 有 20 项任务需要执行. 牵引车在每两点之间移动需 5min, 上下滚装船一次需 15 min, 挂/摘挂车的时间为 5 min. 滚装船可同时容纳 3 台牵引车进行作业. 为更清晰地体现调整过程, 每个模块仅显示任务编号、牵引车编号、牵引车到达和离开滚装船的时间四项. 三个步骤分别为记录初始参数、按上船时间排列形成任务的执行次序、推迟违反舱容约束的任务的上船时间.

合并任务序列

为了减少牵引车的使用数量, 将在时间上不发生冲突的任务序列进行合并. 由于滚装船期分布于整个规划期, 因此部分任务序列的结束时间要早于某些序列的开始时间. 从第一条任务序列开始, 从剩余序列中寻找开始时间在其结束时间之后的序列, 若存在, 从中随机选择一条, 将两条序列合并到一起. 重复此操作, 直至无可合并序列.

本文的初始解生成方式即为现实调度规则.

5.3 邻域搜索

邻域搜索采用以下两种方式, 被选择的概率分别为 δ 和 $1 - \delta$.

随机选择解中的两条任务序列和一个时段, 在两条序列中找到对应于该时段内执行的任务段, 并将二者交换. 此时可能存在三种情况: 1) 选中的两条任务序列中均存在该时段内执行的任务, 直接交换两任务段; 2)

只有一条任务序列中存在该时段执行的任务, 则将该任务段从原任务序列中取出, 插入另一条任务序列的相应位置; 3) 若两条任务序列中均没有在该时段内执行的任务, 则重新进行选择.

当调度方案只有一条任务序列时, 若存在该时段内执行的任务段, 将任务段在任意一点处打断, 前后交换. 若不存在, 重新选择时段进行此操作.

若新植入的任务段的第一项任务开始时间晚于前一项任务的结束时间, 按照时间窗约束将该任务段的开始时间提前至前一任务的结束时间或时间窗较小值, 更新附属矩阵.

随机选择一个任务序列, 将其开始时间在第一项任务时间窗内重新随机生成, 更新附属矩阵并进行调整, 合并任务序列.

6 算例

算例数据根据大连湾新港货滚码头的运营情况获得. 规划期为 0:00~24:00, 每艘滚装船在港停泊 6 h, 其中下船作业和上船作业时间均为 2.5 h, 维护时间为 1 h. T_{∇} 和 T_{Δ} 为 5 min, T_{r-r} 为 15 min. 滚装船舱内同时最多有 3 台牵引车作业.

牵引车使用的固定成本 $f = 200$, 可变成本 $c_1 = 10$, 机会成本 $c_2 = 6$.

算法的初始温度为 1 000, 终止温度为 1, 降温速率为 0.8, 内循环次数为 1 000. 邻域搜索方式选择参数 $\delta = 0.5$.

直接任务的数量分为三个等级, 具体见表 1.

表 1 不同等级的直接任务数量
Table 1 Direct-task quantities of different grades

等级	每艘船下船任务数		每艘船上船任务数	
	到堆场	到仓库	从堆场	从仓库
I	3	2	3	2
II	6	4	6	4
III	12	8	12	8

在港时间无交叉情况下滚装船停留情况如表 2 所示, 每组三个算例的任务数量分别为 I 级, II 级和 III 级. 堆场和仓库之间任务交换的情况如表 3 所示.

表 2 滚装船在港时间无交叉算例设置
Table 2 Numerical examples without overlapping ship port time

算例编号	滚装船数量	滚装船在港时间
1,2,3	1	[6,12]
4,5,6	2	[6,12],[12,18]
7,8,9	3	[6,12],[12,18],[18,24]
10,11,12	4	[0,6],[6,12],[12,18],[18,24]

表 3 船舶在港时间无交叉时的间接任务数
Table 3 Indirect-task quantity without overlapping ship port time

算例编号	堆场→仓库	仓库→堆场
1,2,3	5	5
4,5,6	10	10
7,8,9	15	15
10,11,12	20	20

滚装船在港时间有交叉的算例设置见表 4, 每组三个算例的任务数量仍分别为 I, II 和 III 级. 相应的间接任务数量设置见表 5.

全部试验均在 Intel(R)Core(TM)i7-3770 CPU@3.40 GHz 处理器, 4G 内存的 PC 上运行, 采用 MATLAB R2014a 编程实现. 最长计算时间为 98.18 s, 证明算法具有较高的时间效率.

滚装船在港时间无交叉的 12 个算例求解结果及对比见表 6, 有交叉的 21 个算例的求解结果及对比见表 7.

通过表 6 和表 7 的求解结果可以看出, 由于在目标函数中, 牵引车启用的固定成本起到主导作用, 因此在所有算例中, 本文所设计的 SA 算法均能寻到最少的牵引车数量来完成所有的任务. 部分算例能够在运营时间或者等待时间方面寻到最优值. 另外, 从使用的牵引车数量角度而言, 当前的调度规则存在一定的资源浪费; 从牵引车总的运营时间角度而言, 由于货滚码头区域面积较小, 同时节点数量较少, 牵引车执行完一项任务后转而执行另一项任务的准备时间优化空间较小, 因此三者的差异不大; 从牵引车总的等待时间角度而言, 存在一种大致的趋势, 即等待时间与使用的牵引车台数成正相关, 而且从三者的对比发现, 此部分的优化空间较大.

表 4 滚装船在港时间有交叉算例设置

Table 4 Numerical examples with overlapping ship port time

算例编号	滚装船数量	滚装船在港时间
13,14,15	2	[0,6],[1,7]
16,17,18	3	[6,12]
19,20,21	4	[8,14]
22,23,24	5	[14,20]
25,26,27	6	[15,21]
28,29,30	7	[16,22]
31,32,33	8	[18,24]

表 5 船舶在港时间有交叉时的间接任务数

Table 5 Indirect-task quantity with overlapping ship port time

算例编号	堆场→仓库	仓库→堆场
13,14,15	6	6
16,17,18	11	11
19,20,21	13	13
22,23,24	19	19
25,26,27	20	20
28,29,30	21	21
31,32,33	23	23

注: “滚装船在港时间”一列, 除第一项外, 每一项均为在前面所有项累积的基础上新增的时间窗约束.

表 6 在港时间无交叉情形下的求解结果及对比

Table 6 Results and comparison without overlapping ship port time

算例编号	现有调度规则				下界值			SA				Gap1	Gap2
	a1	b1	c1	d1	LB	LB _V	LB _T	a2	b2	c2	d2		
1	481	2	485	0	275	1	450	282	1	490	0	41.4 %	2.5 %
2	729	3	775	0	525	2	750	549	2	785	185	24.7 %	4.6 %
3	1 246	5	1 350	210	825	3	1 350	844	3	1 350	191	32.3 %	2.3 %
4	564	2	985	0	350	1	900	387	1	985	228	31.4 %	10.6 %
5	862	3	1 570	0	650	2	1 500	686	2	1 580	225	20.4 %	5.5 %
6	1 500	5	2 730	450	1 050	3	2 700	1 113	3	2 790	481	25.8 %	6.0 %
7	648	2	1 485	0	425	1	1 350	445	1	1 470	0	31.3 %	4.7 %
8	994	3	2 365	0	775	2	2 250	809	2	2 320	223	18.6 %	4.4 %
9	1 947	6	4 105	630	1 275	3	4 050	1 326	3	4 095	439	31.9 %	4.0 %
10	931	3	1 985	0	700	2	1 800	750	2	1 985	187	19.4 %	7.1 %
11	1 327	4	3 160	0	1 100	3	3 000	1 138	3	3 125	169	14.2 %	3.5 %
12	2 198	6	5 485	840	1 700	4	5 400	1 755	4	5 405	541	20.2 %	3.2 %

注: a1, a2 为目标函数值; b1, b2 为使用的牵引车台数; c1, c2 为牵引车的总运营时间; d1, d2 为牵引车的总等待时间;

Gap1=(a1-a2)/a1×100 %; Gap2=(a2-LB)/LB×100 %.

而由表 6 和表 7 的结果对比可见, 本文所设计的 SA 算法优化效果非常理想, 在滚装船在港时间无交叉的 12 个算例中, SA 求解的总成本较现有调度规则平均可优化 26.0 %, 与下界值的差距平均为 4.9 %; 在滚装船在港时间有交叉的 21 个算例中, 两个数值分别为 15.6 % 和 3.4 %, 说明运用该算法可极大降低现有调度规则的成本. 同时除算例 4 外, SA 求解结果与下界值的差距均控制在 10 % 以内, 算例 4 中 10.6 % 的差距也处于可接受的范围内. 因此, 本文设计的算法对于解决货滚码头甩挂运输调度问题是有效的. 另外, 通过结果比较还可看出, 当滚装船在港停留时间无交叉时, 算法的优化效果相对更好一些.

7 结束语

货滚甩挂码头区域内部的挂车拖运完全依靠牵引车进行, 这些拖运任务数量多且执行时间严格, 加之滚装船作业舱容的限制和多船同步作业的影响, 牵引车的调度非常关键且难度较大. 然而理论界却没有给予此问题足够的重视, 实际生产中码头则指派部分牵引车在一艘滚装船停泊期间固定服务该滚装船, 导致牵引车等待的情况频发, 造成了极大的资源浪费. 本文创新性地解决了货滚甩挂码头牵引车调度优化问题. 建立了混合整数规划模型, 设计了模拟退火算法, 并推导了运营成本的下界值. 根据实际运营情况设置算例, 将本文算法的求解结果与现有规则调度结果和下界值进行了对比, 证明了本文算法的有效性, 尤其在牵引车数量寻优方面存在较强的优势. 未来的研究将关注滚装船航线两端货流不均衡导致的空挂调运问题, 并考虑上船作业过程中临时到达的挂车对调度计划的实时扰动.

表7 在港时间有交叉情形下的求解结果及对比

Table 7 Results and comparison with with overlapping ship port time

算例编号	现有调度规则				下界值			SA				Gap3	Gap4
	e1	f1	g1	h1	LB*	LB _V *	LB _T	e2	f2	g2	h2		
13	737	3	820	0	530	2	780	545	2	795	125	26.0 %	2.8 %
14	1 233	5	1 400	0	1 030	4	1 380	1 036	4	1 385	49	16.0 %	0.6 %
15	2 665	11	2 540	420	2 030	8	2 580	2 085	8	2 625	470	21.8 %	2.7 %
16	820	3	1 320	0	605	2	1 230	624	2	1 310	57	23.9 %	3.1 %
17	1 366	5	2 195	0	1 155	4	2 130	1 200	4	2 145	425	12.2 %	3.9 %
18	2 916	11	3 920	630	2 255	8	3 930	2 395	8	3 995	1 291	17.9 %	6.2 %
19	886	3	1 700	30	665	2	1 590	691	2	1 700	81	22.0 %	3.9 %
20	1 485	5	2 875	60	1 265	4	2 790	1 326	4	2 790	610	10.7 %	4.8 %
21	3 163	11	5 195	975	2 465	8	5 190	2 550	8	5 210	814	19.4 %	3.4 %
22	976	3	2 240	30	745	2	2 070	787	2	2 250	115	19.4 %	5.6 %
23	1 624	5	3 710	60	1 395	4	3 570	1 425	4	3 630	195	12.3 %	2.2 %
24	2 981	11	3 990	1 155	2 695	8	6 570	2 756	8	6 585	586	7.5 %	2.3 %
25	1 033	3	2 580	30	800	2	2 400	872	2	2 700	220	15.6 %	9.0 %
26	1 730	5	4 345	60	1 500	4	4 200	1 542	4	4 285	278	10.9 %	2.8 %
27	3 642	11	7 835	1 365	2 900	8	7 800	2 952	8	7 890	368	18.9 %	1.8 %
28	1 292	4	2 915	60	1 055	3	2 730	1 061	3	2 765	0	17.9 %	0.6 %
29	2 241	7	4 975	120	2 005	6	4 830	2 064	6	4 945	402	7.9 %	2.9 %
30	4 676	15	9 050	1 680	3 905	12	9 030	4 009	12	9 055	1 002	14.3 %	2.7 %
31	1 561	5	3 295	120	1 315	4	3 090	1 370	4	3 150	453	12.2 %	4.2 %
32	2 967	10	5 655	240	2 515	8	5 490	2 597	8	5 565	691	12.5 %	3.3 %
33	5 509	18	10 314	1 902	4 915	16	10 290	5 008	16	10 305	905	9.1 %	1.9 %

注: e1, e2 为目标函数值; f1, f2 为使用的牵引车台数; g1, g2 为牵引车的总运营时间; h1, h2- 牵引车的总等待时间;

Gap3=(e1-e2)/e1×100%; Gap4=(e2-LB*)/LB*×100%.

参考文献:

- [1] 吕延昌. 鲁辽滚装船甩挂运输的业务流程设计. 物流技术, 2014, 33(9): 36-38.
Lv Y C. Design of business process of ro-ro drop & pull transportation in Shandong-Liaoning region. Logistics Technology, 2014, 33(9): 36-38. (in Chinese)
- [2] Chao I. A tabu search method for the truck and trailer routing problem. Computers & Operations Research, 2002, 29(1): 33-51.
- [3] Lin S, Yu V F, Chou S. Solving the truck and trailer routing problem based on a simulated annealing heuristic. Computers & Operations Research, 2009, 36(5): 1683-1692.
- [4] Lin S, Yu V F, Chou S. A note on the truck and trailer routing problem. Expert Systems with Applications, 2010, 37(1): 899-903.
- [5] Lin S, Yu V F, Lu C. A simulated annealing heuristic for the truck and trailer routing problem with time windows. Expert Systems with Applications, 2011, 38(12): 15244-15252.

- [6] Villegas J G, Prins C, Prodhon C, et al. A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 2011, 38(9): 1319–1334.
- [7] Villegas J G, Prins C, Prodhon C, et al. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 2013, 230(2): 231–244.
- [8] Derigs U, Pullmann M, Vogel U. Truck and trailer routing-problems, heuristics and computational experience. *Computers & Operations Research*, 2013, 40(2): 536–546.
- [9] 胡志华, 陶莎. 基于混合进化算法的甩挂配送问题. *公路交通科技*, 2013, 30(5): 147–152.
Hu Z H, Tao S. Distribution by trailer pick-up transport based on hybrid evolutionary algorithm. *Journal of Highway and Transportation Research and Development*, 2013, 30(5): 147–152. (in Chinese)
- [10] Tan K C, Chew Y H, Lee L H. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 2006, 172: 855–885.
- [11] Lee L H, Tan K C, Ou K, et al. Vehicle capacity planning system: A case study on vehicle routing problem with time windows. *Ieee Transactions on Systems Man and Cybernetics Part A-Systems and Humans*, 2003, 33(2): 169–178.
- [12] Francis P, Zhang G, Smilowitz K. Improved modeling and solution methods for the multi-resource routing problem. *European Journal of Operational Research*, 2007, 180(3): 1045–1059.
- [13] Xue Z, Zhang C, Lin W, et al. A tabu search heuristic for the local container drayage problem under a new operation mode. *Transportation Research Part E-Logistics and Transportation Review*, 2014, 62: 136–150.
- [14] Xue Z, Lin W, Miao L, et al. Local container drayage problem with tractor and trailer operating in separable mode. *Flexible Services and Manufacturing Journal*, 2015, 27(2-3): 431–450.
- [15] Xue Z, Zhang C, Yang P, et al. A combinatorial benders' cuts algorithm for the local container drayage problem. *Mathematical Problems in Engineering*, 2015(2015): 1–7.
- [16] 胡志华, 曹杨, 王云霞. 集装箱集散的空重箱循环甩挂调度方法. *武汉理工大学学报*, 2012, 34(10): 68–73.
Hu Z H, Cao Y, Wang Y X. Cyclic tractor-and-trailer scheduling for consolidation and distribution of loaded and empty containers. *Journal of Wuhan University of Technology*, 2012, 34(10): 68–73. (in Chinese)
- [17] 胡志华. 集装箱码头间互拖的集卡甩挂运输调度问题. *重庆交通大学学报(自然科学版)*, 2013, 32(2): 313–317.
Hu Z H. Scheduling of container truck-and-trailer transport between two ports. *Journal of Chongqing Jiaotong University (Natural Science)*, 2013, 32(2): 313–317. (in Chinese)
- [18] 钟慧玲, 吴聪, 张冠湘, 等. 内河港口甩挂运输组织中场站选址问题研究. *交通运输系统工程与信息*, 2013, 13(3): 126–131.
Zhong H L, Wu C, Zhang G X, et al. Freight station location of inland port semi-trailer swap transport organization. *Journal of Transportation System Engineering and Information Technology*, 2013, 13(3): 126–131. (in Chinese)
- [19] 余莉, 林桦, 陈伯如. 网络型甩挂运输模式下的车辆调度问题. *交通运输工程与信息学报*, 2014, 12(2): 58–64.
Yu L, Lin H, Chen B R. Research of vehicle scheduling problem of the network drop and pull transport. *Journal of Transportation Engineering and Information*, 2014, 12(2): 58–64. (in Chinese)
- [20] 侯玉梅, 贾震环, 田歆, 等. 带软时间窗整车物流配送路径优化研究. *系统工程学报*, 2015, 30(2): 240–250.
Hou Y M, Jia Z H, Tian X, et al. Research on the optimization on the vehicle logistics distribution with soft time windows. *Journal of Systems Engineering*, 2015, 30(2): 240–250. (in Chinese)

作者简介:

杨珍花(1991—), 女, 山东青岛人, 博士生, 研究方向: 物流系统优化与模拟仿真, Email: yangzhenhua91@126.com;

邢磊(1989—), 男, 山东威海人, 博士生, 研究方向: 物流系统优化与模拟仿真, Email: xinglei0915@163.com;

徐奇(1984—), 女, 广西桂林人, 副教授, 研究方向: 物流系统优化与模拟仿真, Email: cogi@163.com;

靳志宏(1963—), 男, 辽宁沈阳人, 博士生导师, 教授, 研究方向: 物流系统优化与模拟仿真, Email: jinzhihong@dlmu.edu.cn.