

支线航运网络下集装箱分配与船舶调度联合优化

孔灵睿, 计明军*, 孙以宁, 关云潇, 郭兴海

(大连海事大学交通运输工程学院, 辽宁 大连 116026)

摘要: 将产生集装箱运输任务的货源点加入至轴辐式航运网络的支线运输网络中, 构建了三级支线运输网络. 考虑货源点集装箱在不同喂给港的分配对支线船舶运输成本的影响, 以三级支线网络的总运输成本最小为目标, 建立了集装箱分配与支线船舶调度联合优化模型. 针对模型, 首先设计两阶段算法进行求解; 然后, 基于列生成的思想设计了整合优化求解算法. 算例实验表明对于小规模算例, 整合优化算法的求解结果与 Gurobi 的计算结果仅相差 1% 左右; 对于大规模算例, 整合优化算法的求解结果均优于 Gurobi, 从而验证了整合优化算法的效率; 此外, 整合优化算法的求解结果相比于两阶段算法有显著提高, 证明了整合优化集装箱分配与支线船舶调度的重要性.

关键词: 支线集装箱船舶调度; 集装箱分配; 整合优化; 列生成

中图分类号: TP273

文献标识码: A

文章编号: 1000-5781(2022)06-0811-16

doi: 10. 13383/j.cnki.jse.2022.06.007

Integrated optimization for container allocation and containership routing under the feeder shipping network

Kong Lingrui, Ji Mingjun*, Sun Yining, Guan Yunxiao, Guo Xinghai

(Institute of Transportation Engineering, Dalian Maritime University, Dalian 116026, China)

Abstract: Cargo points that generate container transportation tasks are added into the feeder network of the hub & spoke shipping network to construct a three-level feeder transportation network. Considering the impact of the distribution of containers among different feeder ports on the feeder transportation cost, an integrated optimization model for container allocation and feeder routing is established to minimize the total transportation cost. First, a two-phase algorithm is designed. Then, an integrated optimization algorithm based on the method of column generation is developed. Experimental results show that for small-scale instances, the difference between the results of the integrated optimization algorithm and Gurobi is only about 1%, for large-scale instances, the integrated optimization algorithm is superior to Gurobi, which confirms the efficiency of the algorithm. In addition, compared with the two-phase algorithm, the results of the integrated optimization algorithm are significantly improved, which verifies the importance of the integrated optimization.

Key words: feeder containership routing; container allocation; integrated optimization; column generation

1 引言

在经济全球化的影响下, 海运贸易量不断增长, 集装箱运输行业在过去的数十年间呈快速、稳定的发展趋势. 2018 年, 全球集装箱海运量已达到 2.01 亿 TEU. 集装箱海运业的强劲发展, 有力推动了集装箱船舶的

收稿日期: 2019-10-31; 修订日期: 2021-01-28.

基金项目: 国家自然科学基金资助项目(71572022; 91071035).

*通信作者

大型化. 目前, 全球最大集装箱船舶的运载能力已超过 20 000 TEU. 然而大型集装箱船舶在为船公司带来规模经济效益的同时, 其靠港的吃水要求也决定了大型集装箱船舶只能挂靠在吃水较深的干线港. 而仅靠干线港之间的干线运输难以满足全方位的现代化运输需求, 国际集装箱海运格局逐步由原来单一的港到港的运输向着干线运输和支线运输相结合的方向衍化, 形成了以枢纽港(干线港)为转运中心的轴辐式航运网络. 以枢纽港口为中心的轴辐式航运网络, 是海运一体化进程中出现的一种能够适应全球海运系统的网络形态^[1], Msakni 等也通过实例验证了应用轴辐式航运网络的重要意义^[2]. 在轴辐式航运网络中, 大船通常被用来服务于各大枢纽港之间的干线运输以获得规模经济效益, 小船则被用来完成支线运输任务以满足枢纽港所对应若干喂给港(支线港)的运输需求^[3]. 对于轴辐式航运网络, 需要明确枢纽港的位置、喂给港的挂靠方案以及枢纽港之间的班轮运输航线和服务时间表. 在此基础上, 如何规划最优的支线集装箱船舶调度方案, 实现支线运输成本的最小化, 就成为了船公司日常需要考虑的重要问题.

支线集装箱船舶运输网络同车辆路径问题(vehicle routing problem, VRP)的网络相似, 均是包含一个配送中心(枢纽港)以及若干顾客点(喂给港)的二级运输网络, 如图 1 虚框内部分所示.

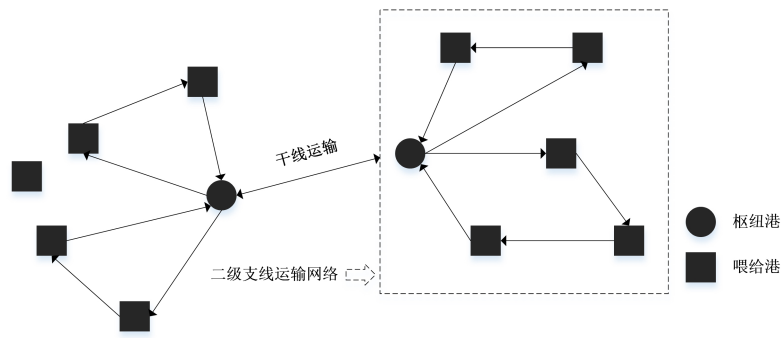


图 1 二级支线运输网络

Fig. 1 The two-level feeder transportation network

从现代供应链和产业链的角度看, 海上集装箱运输仅是其中的一个环节, 只能满足客户的部分需求. 2016 年起, 以马士基集团与法国达飞轮船为首的集装箱航运巨头试图将供应链整合战略应用到海洋运输服务中. 尽管战略执行有所不同, 但均以加强海洋与内陆腹地的连接、接近海洋服务终端客户为目标, 以期深化企业与用户合作, 打造共赢、共存和良性发展的供应链生态圈. 为顺应这一发展趋势, 本文将产生集装箱运输任务的货源点加入至轴辐式航运网络的支线运输网络中, 构建三级支线运输网络, 如图 2 所示.

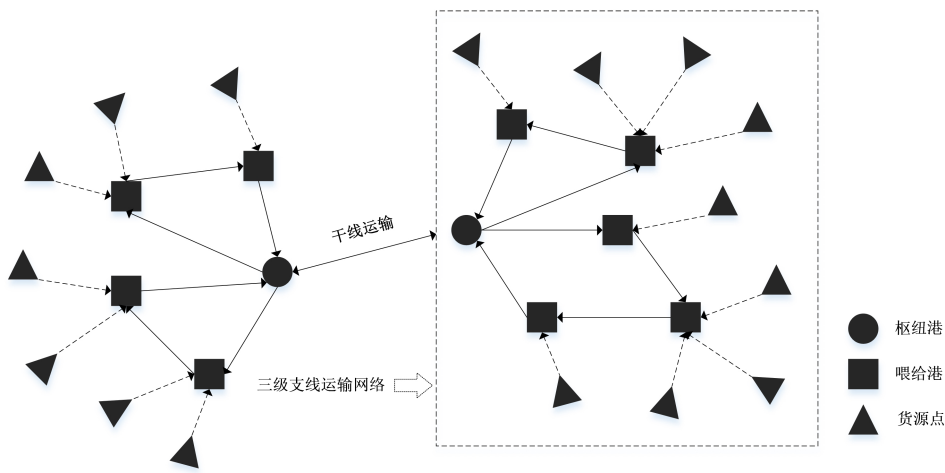


图 2 三级支线运输网络

Fig. 2 The three-level feeder transportation network

不同于传统的支线运输网络, 三级支线运输网络不仅需要对支线集装箱船舶调度方案进行决策, 还需同

时考虑各货源点的集装箱到喂给港的分配方案. 即货源点的集装箱并非提前分配至相应的喂给港, 而是同支线集装箱船舶调度计划进行联合优化. 若一个货源点的集装箱通过不同的喂给港进行运输, 最优的支线船舶调度方案可能会发生很大的变化, 最小的运输成本也将随之改变. 通过构建三级支线网络并对其进行联合优化研究, 能够为航运企业向供应链终端整合提供思路, 同时可以降低整体支线网络的总运输成本.

以往有关轴辐式航运网络的研究主要集中在枢纽港之间的干线运输上^[4-7], 有关支线船舶调度的研究较少. 关于支线船舶调度问题, Sambrocos 等^[8]将其看作是车辆路径问题(VRP)的一个扩展, 认为支线船舶调度是从枢纽港出发, 向喂给港运货的具有容量约束的 VRP; 靳志宏等^[9]同样将 VRP 理论应用到支线集装箱船舶调度问题中, 在现有研究的基础上, 加入对船舶容量约束与时间窗约束的考虑, 构建了混合整数优化模型, 并采用粒子群算法进行求解; Karlafti 等^[10]考虑了更多的实际情况, 研究了喂给港既有卸箱需求也有装箱需求的情况下, 枢纽港与众多喂给港之间的支线集装箱船舶调度问题, 并采用混合遗传算法进行求解; 为探究船型的选择对于支线集装箱船舶运输成本的影响, Ji 等^[11]构建了多船型的支线集装箱船舶调度优化模型, 并设计改进遗传算法进行求解; 考虑船舶航速对于支线船舶调度方案的影响, Santini^[12]等, 计明军等^[13]研究了可变航速下的支线船舶调度问题. 然而, 以往的研究均是基于二级支线网络, 假定各喂给港的装箱量已知, 即各个货源点的集装箱已提前分配至相应的喂给港, 再对支线集装箱船舶调度方案进行优化. 缺乏对于供应链终端的货源点的考虑, 因此难以适应航运企业向供应链终端进行整合的发展趋势, 也无法使整体支线运输网络的总成本达到最小.

而对于车辆路径问题, 也有学者将其拓展为三级运输网络. 其中与本研究最为相似的是考虑需求分配的车辆路径问题(vehicle routing with demand allocation problem, VRDAP). 该问题最早由 Ghoniem 等^[14]提出, 应用于非营利性的食物配送方面, 以最小化运输距离为目标建立了混合整数规划模型并设计了精确求解算法进行求解; Solak 等^[15]以及 Reihaneh 等^[16]随后也分别设计了更优的精确求解算法对 VRDAP 进行求解. 本文研究的实质可以看成是 VRDAP 在支线航运网络的应用, 但又与 VRDAP 研究有着较大的区别: VRDAP 研究假设所有的车辆具有相同的运载能力, 而本文的三级支线运输网络中包含两种运载工具, 集装箱运输船舶与集装箱运输车辆, 且在实际运输过程中存在着不同大小的支线集装箱船舶, 因此需要对船舶不同的运载能力进行考虑; VRDAP 研究仅考虑了配送问题, 而支线集装箱船舶运输既需考虑卸船需求也需考虑装船需求; VRDAP 研究没有考虑任何时间上的约束, 而相比于陆路运输节点, 集装箱航运港口具有更为严格的时间要求, 包含喂给港的作业时间窗以及枢纽港的运输时刻表. 从研究内容以及约束条件上看, 本文的研究都更为复杂, 更贴合支线集装箱船舶运输的实际情况.

基于上述考虑, 本文将货源点加入至传统的支线运输网络中, 将传统的二级支线运输网络拓展成三级支线运输网络. 针对支线集装箱船舶运输的特点, 考虑船舶容量、喂给港时间窗以及枢纽港运输班期等现实因素, 分析货源点集装箱在喂给港的分配与支线船舶调度计划之间的内在联系, 对集装箱分配方案以及支线集装箱船舶调度计划进行联合优化, 以期能够降低整个三级支线运输网络的总成本, 为航运企业的升级转型提供有益借鉴.

2 问题描述与数学模型

如图 3 所示, 本文研究的三级支线运输网络包括一个枢纽港、挂靠在该枢纽港下的喂给港以及若干产生集装箱运输任务的货源点. 通过对比图 3 中的两个不同的运输方案可知, 若某一货源点的集装箱被分配至另一个不同的喂给港, 那么最终的支线船舶调度方案可能会发生改变, 运输成本也会随之发生变化. 若提前确定好货源点的集装箱到喂给港的分配, 并不能够使得整个三级支线网络的总成本达到最小. 因此需要对集装箱分配以及支线船舶调度进行联合优化, 以最小化三级支线网络的总运输成本.

在本文所描述的网络中, 从枢纽港到每个喂给港的卸箱量已知, 每个喂给港的装箱量则是分配至该喂给港的集装箱量的总和. 若某一喂给港既没有卸载集装箱, 也没有任何一个货源点的集装箱被分配至该喂给

港,则不需要任何一艘集装箱船舶挂靠至该喂给港.本文不但考虑船舶到达各喂给港的时间窗约束,也考虑所有货源点的集装箱必须在规定时间内之前送至枢纽港的班期约束.此外,本文考虑多种船型的集装箱船舶,不同类型的集装箱船舶具有不同的容量、经济航速与运输费用.

本文需要解决的问题包括为每个货源点的集装箱选择装船的喂给港、为有集装箱运输需求的喂给港分配指定船舶并确定每艘船舶的运输路线,以满足运输需求、船舶容量及时间窗限制,并使得船舶运输与集装箱分配(集装箱从货源点至喂给港的运输成本)的总成本最小.

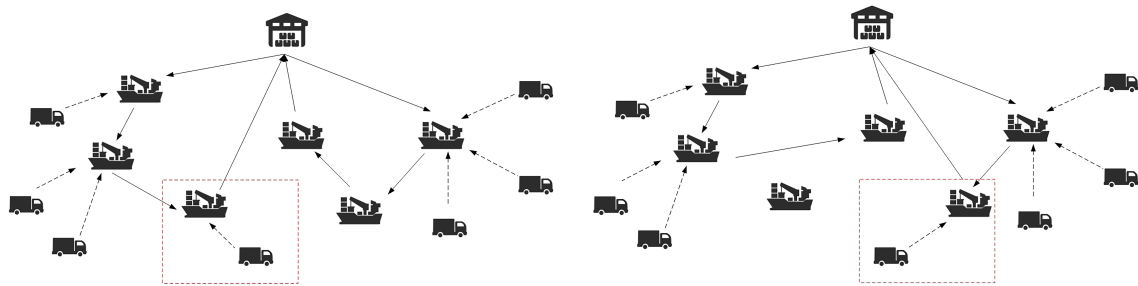


图3 货源点集装箱在不同喂给港的分配对支线船舶调度方案的影响

Fig. 3 The impact of the container distribution among different feeder ports on the feeder routing plan

表1 参数与变量说明

Table 1 Description of parameters and variables

符号表示	符号含义
集合	
Ω	喂给港集合
K	可用船舶集合
C	产生集装箱运输任务的货源点集合
参数	
O	枢纽港
O'	虚拟枢纽港,即所有支线船舶的终点
Z	一个充分大的正数
a_i	喂给港 i 最早可以停靠的时间
b_i	喂给港 i 最晚可以停靠的时间
d_{ij}	$i-j$ 段的距离
v_k	船舶 k 的经济航速
Q_k	船舶 k 的容量
p_i	需要从枢纽港运输至喂给港 i 的集装箱量(喂给港 i 的卸箱量)
w_c	货源点 c 的运输任务量(集装箱量, TEU)
st_i	港口 i 的装卸效率(h/TEU)
l_c	货源点 c 的集装箱运至枢纽港的最晚时间要求
f_{kij}	船舶 k 从港口 i 航行至港口 j 的成本,与船型以及港口之间的距离有关
e_{ci}	货源点 c 的集装箱运输到港口 i 的成本
变量	
x_{kij}	0-1 决策变量,船舶 k 从港口 i 直接行驶至港口 j ,则为 1,否则为 0
y_{ci}	0-1 决策变量,若货源点 c 的集装箱通过喂给港 i 运输,则为 1,否则为 0
g_i	0-1 决策变量,若喂给港 i 有运输需求,则为 1,否则为 0
t_{ki}	集装箱船舶 k 到达 i 港的时间
s_{ki}	集装箱船舶 k 在 i 港的作业时间
u_{ki}	集装箱船舶 k 离开港口 i 时的载箱量
q_i	在港口 i 需要进行装船运输的集装箱量

2.1 模型假设

基于支线集装箱船舶运输的特点,提出如下假设:

1) 任意喂给港最多只能由一艘船舶挂靠;

2) 所有的支线集装箱船舶均从枢纽港出发, 完成相应的装卸任务后, 最终返回枢纽港;

3) 在喂给港卸下的集装箱全部来自枢纽港, 且在喂给港装船的集装箱只能运输至枢纽港, 即喂给港之间不存在集装箱流量;

4) 所有货源点的集装箱均不能被拆分, 即某一货源点的集装箱必须全部运输至同一喂给港; 若同一位置的集装箱允许被运往不同的喂给港, 则被视为不同的货源点;

5) 在调度的过程中需要保证每一个货源点的运输需求都被满足.

2.2 模型构建

为了构建数学模型, 定义如表 1 所示的参数和变量.

基于表 1 中的参数以及变量定义, 集装箱分配与多船型支线集装箱船舶调度联合优化模型为

$$\text{Min} \sum_{k \in K} \sum_{i \in (\Omega \cup O)} \sum_{j \in (\Omega \cup O')} x_{kij} f_{kij} + \sum_{c \in C} \sum_{i \in \Omega} y_{ci} e_{ci}, \quad (1)$$

s.t.

$$\frac{1}{Z} \left(\sum_{c \in C} y_{ci} + p_i \right) \leq g_i \leq \sum_{c \in C} y_{ci} + p_i, \quad \forall i \in \Omega, \quad (2)$$

$$\sum_{k \in K} \sum_{i \in (\Omega \cup O \setminus j)} x_{kij} = g_j, \quad \forall j \in \Omega, \quad (3)$$

$$\sum_{j \in (\Omega \cup O')} x_{kOj} = 1, \quad \forall k \in K, \quad (4)$$

$$\sum_{i \in (\Omega \cup O \setminus j)} x_{kij} = \sum_{h \in (\Omega \cup O' \setminus j)} x_{kjh}, \quad \forall j \in \Omega, k \in K, \quad (5)$$

$$t_{ki} + s_{ki} + \frac{d_{ij}}{v_k} - t_{kj} \leq Z(1 - x_{kij}), \quad \forall i \in (\Omega \cup O), \forall j \in (\Omega \cup O' \setminus i), k \in K, \quad (6)$$

$$s_{kO} = \sum_{i \in (\Omega \cup O)} \sum_{j \in \Omega \setminus i} x_{kij} p_j \text{st}_O, \quad \forall k \in K, \quad (7)$$

$$s_{ki} = \text{st}_i(p_i + q_i), \quad \forall i \in \Omega, k \in K, \quad (8)$$

$$a_i \leq t_{ki} \leq b_i, \quad \forall i \in \Omega, k \in K, \quad (9)$$

$$t_{kO'} \leq l_c + Z \left(1 - \sum_{i \in \Omega \cup O \setminus j} x_{kij} \right) + Z(1 - y_{cj}), \quad \forall k \in K, j \in \Omega, c \in C, \quad (10)$$

$$u_{kO} = \sum_{i \in \Omega \cup O} \sum_{j \in (\Omega \setminus i)} x_{kij} p_j, \quad \forall k \in K, \quad (11)$$

$$u_{kj} \geq u_{ki} - p_j + q_j - Z(1 - x_{kij}), \quad \forall i \in (\Omega \cup O), j \in (\Omega \cup O' \setminus i), k \in K, \quad (12)$$

$$u_{ki} \leq Q_k, \quad \forall i \in (\Omega \cup O), k \in K, \quad (13)$$

$$q_i = \sum_{c \in C} y_{ci} w_c, \quad \forall i \in \Omega, \quad (14)$$

$$\sum_{i \in \Omega} y_{ci} = 1, \quad \forall c \in C, \quad (15)$$

$$x, y, g \in \{0, 1\} \quad t, s, u, q \geq 0,$$

其中式(1)中的目标函数表示最小化船舶运输和集装箱分配的总成本, 即最小化三级支线网络的总运输成本; 约束(2)定义了变量 g_i , 若喂给港 i 有卸船集装箱, 或至少有一个货源点被分配至喂给港 i , 则喂给港 i 有运输需求, $g_i = 1$, 否则, $g_i = 0$; 约束(3)表示每个有运输需求的喂给港有且仅有一艘集装箱船舶挂靠, 若某一喂给港没有运输需求, 则该喂给港没有集装箱船进行挂靠; 约束(4)规定所有集装箱船舶均从枢纽港出发, 其中若 $x_{kOO'} = 1$, 则表示船舶 k 没有被实际投入使用, 是一条虚拟航线; 约束(5)可以保证运输流的平衡, 即

驶入某一喂给港的集装箱船舶必须从该喂给港驶出;约束(6)定义了船舶从一个港口到达下一个港口的时间,保证了船舶航行的连续性,且能够消除子回路;约束(4)~约束(6)相结合可以保证每艘集装箱船舶仅分配一条运输航线,且最终均返回枢纽港;约束(7)定义了集装箱船在枢纽港进行装船作业的时间;约束(8)定义了集装箱船舶在喂给港进行装卸作业的时间;约束(9)保证船舶到达喂给港的时间满足喂给港的时间窗要求;约束(10)保证船舶返回枢纽港的时间满足船上所有集装箱必须运至枢纽港的最晚时间要求;约束(11)定义了船舶在枢纽港的装箱量;约束(12)保证喂给港的运输需求全部被满足(分配至某一喂给港的货源点的集装箱在该喂给港全部被装船);约束(13)表示船舶的容量约束;约束(14)定义喂给港的装箱量;约束(15)保证每一个货源点均被分配至一个喂给港。

假设货源点的数量为 M , 包含枢纽港在内的港口的数量为 N , 可以选择的船型的数量为 H . 对于一个喂给港来说, 其紧前喂给港有 $N - 1$ 种可能, 其紧后喂给港的选择也有 $N - 1$ 种可能, 可选的船型数量为 H , 那么该喂给港共可能生成 $H(N - 1)^2$ 种方案; 对于 $N - 1$ 个喂给港, 最多可产生 $H(N - 1)^3$ 种方案; 对于货源点的分配共可能产生 $M(N - 1)$ 种方案, 因此问题的复杂度最终可表示为 $MH(N - 1)^4$. 随着喂给港以及货源点数量的增加, 使用线性求解器对模型(1)进行直接求解会消耗大量的时间, 因此在本文的第3节设计了求解算法对大规模问题进行求解。

2.3 两阶段求解模型与算法

为了验证整合优化集装箱分配与支线船舶调度的优势与必要性, 在本节中介绍两阶段求解模型与算法(两阶段求解算法), 用来作为第3节中所提出的整合优化求解算法的对比基础. 基于本文所研究问题的特性, 可以很容易的将问题拆分成第一阶段的集装箱分配问题以及第二阶段的支线集装箱船舶调度问题. 两阶段求解算法描述如下:

步骤1 集装箱分配

在第一步进行集装箱的分配, 目标是最小化总分配成本. 此外, 为了能够得到较好的最终解, 在第一步分配集装箱时也一定程度的考虑船舶运输成本. 由于此时船舶的具体行驶路径是未知的, 船舶运输成本是通过计算船舶从枢纽港到各个有运输需求的喂给港的总运输成本来进行粗略的估计. 因此第一阶段的目标函数可以表示为最小化集装箱分配成本与粗略估计的船舶运输成本的加权平均成本, 如式(16)中的目标函数所示(在集装箱分配模型中, γ 为 0-1 之间的小数, 表示集装箱分配成本在目标函数中所占的比重; 模型中涉及到的其他符号的表示同模型(1)).

$$\text{Min } \gamma \sum_{c \in C} \sum_{i \in \Omega} e_{ci} y_{ci} + (1 - \gamma) \sum_{i \in \Omega} f_{10i} g_i, \quad (16)$$

s.t.

$$\sum_{i \in \Omega} y_{ci} = 1, \forall c \in C, \quad (17)$$

$$\sum_{c \in C} y_{ci} w_c \leq \max_{k \in K} \{Q_1, Q_2, \dots, Q_k\}, \forall i \in \Omega, \quad (18)$$

$$\frac{1}{Z} \left(\sum_{c \in C} y_{ci} + p_i \right) \leq g_i \leq \sum_{c \in C} y_{ci} + p_i, \forall i \in \Omega, \quad (19)$$

$$y, g \in \{0, 1\},$$

其中约束(17)保证每个货源点都被分配且仅分配至一个喂给港; 约束(18)能够保证得到相应的集装箱分配方案后, 在第二阶段进行船舶调度的过程中不会超过船舶的容量约束; 约束(19)定义了变量 g_i .

该模型可直接使用线性求解器进行求解。

步骤2 支线集装箱船舶调度

当每个货源点的集装箱都被分配至相应的喂给港后, 原问题就变成了已知集装箱分配以及各个喂给港运输需求的支线集装箱船舶调度问题. 两阶段算法第二阶段的支线集装箱船舶调度模型与模型(1)结构相

同, 不同点为在支线集装箱船舶调度模型中, 模型(1)中的变量 g_i 和 y_{ci} 成为了已知参数, 为两阶段算法中第一阶段的求解结果. 关于此阶段模型的求解算法并非本文研究重点, 采用以往研究中已提出的启发式算法进行求解. 由于本文考虑的是多船型下的支线集装箱船舶调度问题, 因此参考文献[11]中描述的遗传算法对两阶段算法的第二阶段进行求解.

3 基于列生成的整合优化求解算法

3.1 列生成求解算法原理及框架

列生成算法是求解大规模整数规划问题的优化算法, 其理论是由 Danzig 等^[17]提出的 Dantzig-Wolfe 分解. 目前, 该算法被用于对 VRP 进行求解, 并取得了很好的效果^[18-19]. 考虑到本文所研究的问题为 VRP 的一个变形与拓展, 因此参考求解 VRP 的列生成算法, 基于本文所研究问题的特性, 设计基于列生成的求解算法对问题进行求解.

应用列生成算法, 首先是根据分解原理将模型(1)进行分解变型, 生成与模型(1)等价的, 拥有同样最优解的集分割模型. 在集分割模型中, 用单一加权约束代替了多面体集约束大大减少了约束的数目, 但因这个多面体的顶点可能非常多, 因此增加了约束矩阵的列数. 而这种多列的问题则可以采用列生成技术进行求解.

对于集分割模型, 首先将其线性松弛并将松弛后的集分割模型称为主问题(master problem, MP)模型. 若 n 代表 MP 模型的变量数量, m 表示 MP 模型的约束数量, 根据集分割模型的特点可知 n 远远大于 m . 因此 MP 模型的基本可行解的数量可表示为 C_n^m , 即其可行域的顶点有 C_n^m 个. 当 n 很大时, 即使 m 较小, 总极点数量也十分的庞大. 因此列生成算法的基本思想为: 使用主问题模型所有列的部分子集构建受限主问题(restricted master problem, RMP)模型, 并对 RMP 模型进行求解, 然后通过迭代向 RMP 模型中加入能够使得 RMP 模型的目标函数值继续降低的列(最小化问题), 直至无法找到这样的列时, 即求得 MP 模型的最优解. 实际上, 在线性规划的最优解里, 基变量的数目永远不会超过约束数目. 因此可以不用在求解时生成全部的列, 只是在需要的时候才生成新的列添入至模型当中, 这样就可以使用较少的列获得 MP 模型的最优解, 能够大大加快求解的速度.

新加入列的原则是基于求解线性规划的单纯型法原理: 在列生成算法中, 将 RMP 模型中未加入的列默认为 MP 模型的非基变量. 如在未加入的列中存在某一系列 s , 其检验数小于 0 (最小化问题), 那么列 s 的加入会使得 RMP 模型的目标函数值进一步降低, 因此需要将列 s 加入至 RMP 模型中再次求解 RMP 模型. 这个过程反复进行直至不存在检验数小于 0 的列为止, 则说明 MP 问题已求得最优解. 由于 MP 模型的列十分的多, 不可能通过枚举方法来找到可加入的列, 因此要构造一个子问题(sub-problem, SP)模型, 来获得可加入的列并判断 MP 模型的最优性. 典型的, 通过这种交互方法, 只需要进行有限次迭代即可获 MP 模型的最优解.

在求得 MP 模型的最优解后, 将 0-1 整数变量约束加入至当前的 RMP 模型即可求得集分割模型的整数解. 本文所设计基于列生成的整合优化求解算法的具体步骤总结如下:

步骤 1 构建集分割模型, 并松弛其整数约束以构造 MP 模型(见 3.2 节);

步骤 2 以两阶段算法最终求得的集装箱分配以及船舶调度方案作为列生成算法的初始可行解(即初始列);

步骤 3 用初始列构建 RMP 模型(一条可行的船舶路径及相应的集装箱分配方案对应着 RMP 模型的一列);

步骤 4 构建能为 RMP 模型生成新列, 且能够判断 MP 模型最优性的 SP 模型(见 3.3 节);

步骤 5 用线性求解器 Gurobi 求解 RMP 模型, 获得当前 RMP 模型的对偶变量值;

步骤 6 将当前的对偶变量值代入 SP 模型, 求解 SP 模型, 如果没有可添加的列(任何未加入列的加入均无法使 RMP 模型获得更小的目标函数), 则 MP 模型已求得最优解, 转入步骤 7. 若存在可添加的列使

得 RMP 模型的目标函数值能够进一步降低, 则将相应列加入至 RMP 模型中, 转入步骤 5;

步骤 7 向当前的 RMP 模型中加入 0-1 整数变量约束, 求得集分割模型的整数解.

3.2 集分割模型

集分割模型的部分参数定义如表 2 所示, 其余参数同模型(1).

表 2 参数与变量说明
Table 2 Description of parameters and variables

符号表示	符号含义
参数	
R	所有可行方案的集合, 一个可行方案指一条可行的船舶路径以及该路径下相应的集装箱分配计划
l_r	方案 r 的成本
m_{rc}	表示方案 r 是否运输货源点 c 的集装箱, 是则为 1, 否则为 0, 为相应方案下的已知量
h_{ri}	表示方案 r 是否服务喂给港 i , 是则为 1, 否则为 0, 为相应方案下的已知量
变量	
z_r	0-1 决策变量, 表示方案 r 是否被采用, 是则为 1, 否则为 0

$$\text{Min} \sum_{r \in R} z_r l_r, \quad (20)$$

s.t.

$$\sum_{r \in R} m_{rc} z_r = 1, \forall c \in C, \quad (21)$$

$$\begin{cases} \sum_{r \in R} h_{ri} z_r \leq 1, \forall i \in \Omega, p_i = 0 \\ \sum_{r \in R} h_{ri} z_r = 1, \forall i \in \Omega, p_i > 0, \\ z_r \in \{0, 1\}, \end{cases} \quad (22)$$

其中模型(20)的目标函数表示最小化方案的总成本, 与模型(1)的目标函数等价; 式(21)表示每一个货源点均被分配且仅被分配一次; 式(22)表示若 $p_i = 0$, 则喂给港 i 最多被一条船舶路径访问, 若 $p_i > 0$, 即有从枢纽港运输至喂给港 i 的集装箱, 则该喂给港必须被访问且仅被访问一次.

至此, 模型(1)被转化为与其等价的集分割模型. 模型(1)原本有大量的约束, 而集分割模型的约束仅为 $|C| + |\Omega|$ 个, 与模型(1)相比大为减少. 在集分割模型中, 每一个列都代表着一个可行的方案.

为应用列生成算法, 首先将集分割模型进行线性松弛, 将 $z_r \in \{0, 1\}$ 替换为 $0 \leq z_r \leq 1$, 并称线性松弛后的集分割模型为主问题模型(MP 模型).

3.3 子问题模型

根据单纯型法的原理, 把检验数小于 0 的列代入至 RMP 模型中迭代, 能够优化当前最优解. 因此可将最小化检验数的值作为子问题的目标函数, 若子问题的目标函数值小于零, 则把所求得方案转化成列代入至 RMP 模型中继续求解, 若子问题的目标函数值大于等于 0, 则证明 MP 模型已经求得了最优.

为了表示 MP 模型某一列的检验数, 用 β_c 表示约束(21)所对应的对偶变量, 用 α_i 表示约束(22)所对应的对偶变量. 则 MP 模型某一列的检验数可以表示为

$$z_r - \sum_{c \in C} \beta_c m_{rc} - \sum_{i \in \Omega} \alpha_i h_{ri}. \quad (23)$$

为构建 SP 模型, 重新定义模型(1)中的一些参数, V 表示当前所选船舶的速度, Q 表示当前所选船舶的容量, f_{ij} 表示当前所选船舶在 $i-j$ 段的运输成本.

x_{ij} 为 0-1 决策变量, 表示船舶是否行驶 $i-j$ 段; y_{ci} 的表示同模型(1); t_i 表示船舶到达港 i 的时间; s_i 表示船舶在港 i 的作业时间; u_i 表示船舶离开港口 i 时的载箱量; q_i 表示船舶需要在港口 i 装载的集装箱

总量.

根据定义可知

$$h_{ri} = \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij}, m_{rc} = \sum_{i \in \Omega} y_{ci}, z_r = \sum_{i \in (\Omega \cup O)} \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij} f_{ij} + \sum_{c \in C} \sum_{i \in \Omega} y_{ci} e_{ci},$$

因此式(23)等价于式(24).

$$\sum_{i \in (\Omega \cup O)} \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij} f_{ij} + \sum_{c \in C} \sum_{i \in \Omega} y_{ci} e_{ci} - \sum_{i \in \Omega} a_i \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij} - \sum_{c \in C} \beta_c \sum_{i \in \Omega} y_{ci}. \quad (24)$$

基于以上定义, 有如下模型

$$\text{Min} \sum_{i \in (\Omega \cup O)} \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij} f_{ij} + \sum_{c \in C} \sum_{i \in \Omega} y_{ci} e_{ci} - \sum_{i \in \Omega} a_i \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij} - \sum_{c \in C} \beta_c \sum_{i \in \Omega} y_{ci}, \quad (25)$$

s.t.

$$\sum_{j \in (\Omega \cup O')} x_{Oj} = 1, \quad (26)$$

$$\sum_{i \in (\Omega \cup O \setminus j)} x_{ij} = \sum_{h \in (\Omega \cup O' \setminus j)} x_{jh}, \forall j \in \Omega, \quad (27)$$

$$t_i + s_i + \frac{d_{ij}}{V} - t_j \leq Z(1 - x_{ij}), \forall i \in (\Omega \cup O), \forall j \in (\Omega \cup O' \setminus i), \quad (28)$$

$$s_O = \sum_{i \in (\Omega \cup O)} \sum_{j \in \Omega \setminus i} x_{ij} p_j \text{st}_O, \quad (29)$$

$$s_i = \text{st}_i(p_i + q_i), \forall i \in \Omega, \quad (30)$$

$$a_i \leq t_i \leq b_i, \forall i \in \Omega, \quad (31)$$

$$t_{O'} \leq l_c + Z \left(1 - \sum_{i \in \Omega \cup O \setminus j} x_{ij} \right) + Z(1 - y_{cj}), j \in \Omega, c \in C, \quad (32)$$

$$u_O = \sum_{i \in \Omega \cup O} \sum_{j \in (\Omega \setminus i)} x_{ij} p_j, \quad (33)$$

$$u_j \geq u_i - p_j + q_j - Z(1 - x_{ij}), \forall i \in (\Omega \cup O), \forall j \in (\Omega \cup O' \setminus i), \quad (34)$$

$$u_i \leq Q, \forall i \in (\Omega \cup O), \quad (35)$$

$$q_i = \sum_{c \in C} y_{ci} w_c, \forall i \in \Omega, \quad (36)$$

$$\sum_{i \in \Omega} y_{ci} \leq 1, \forall c \in C, \quad (37)$$

$$\sum_{c \in C} y_{ci} \leq Z \sum_{j \in (\Omega \cup O' \setminus i)} x_{ij}, \forall i \in \Omega, \quad (38)$$

$$x, y \in \{0, 1\}, t, s, u, q \geq 0,$$

其中模型(25)的目标函数表示最小化检验数的值; 式(38)描述了变量 x_{ij} 和变量 y_{ci} 的关系: 若当前船舶路径不经过喂给港 i , 则当前方案下任一货源点都不能分配至喂给港 i ; 式(26)~式(37)的表示可参考式(4)~式(15), 此处不再赘述. 区别为 SP 模型仅生成一条船舶路径及该路径下的集装箱分配方案, 且不用保证每个有运输需求的喂给港都被访问, 也不用保证每个货源点均被分配, 只是求得一条能够使得目标函数最小的可行路径以及相应路径下的集装箱分配方案. 求解 SP 模型, 得到变量 x_{ij} 和变量 y_{ci} 的值之后, 通过转换即可到一个可行方案 r , 及相应的 h_{ri} , m_{rc} 和 l_r 的值.

对于 SP 模型, 即使集装箱被提前分配好了, 该问题可看作是一个 TSP 问题, 用线性求解器求解中大型规模算例仍然较为困难, 因此在本文的 3.4 节中设计了启发式规则对 SP 模型进行求解.

3.4 ALNS 算法求解子问题

子问题是具有容量和时间窗约束的基本最短路径问题的一个扩展,可以定义在一个有向图 $G = (P \cup C, E \cup E')$ 上. 顶点集合由集合 P 和集合 C 构成,其中 P 是港口集合($P = \Omega \cup O \cup O'$), C 为货源点集合;边集合包括集合 E 和集合 E' ,其中 $E = \{(i, j) | i \neq j \in P\}$, $E' = \{(c, i) | c \in C, i \in \Omega\}$. 根据子问题的目标函数,属于集合 E 的边的费用和属于集合 E' 的边的费用分别为

$$w_{ij} = f_{ij} - \alpha_i, \quad (39)$$

$$w_{ci} = e_{ci} - \beta_c. \quad (40)$$

在每次求解主问题得到新的对偶变量值之后都需要对边的费用进行更新. 对边的重新定价,可能会产生负的费用值. Feillet 等^[20]运用动态规划算法对含负费用边的最短路径问题进行求解.若使用动态规划算法求解子问题,在拓展标签的过程中,设某港口有 10 个可选择的货源点,其所能形成的集合为 2^{10} 个,那么当前港口可以拓展的标签数量最多可达 2^{10} . 尽管有很多标签在拓展的过程中由于不可行或是被另一个标签统治而删去,可拓展标签的数量仍然十分多.经实验验证发现通过动态规划算法求解子问题的效率很低.因此本文设计启发式规则对子问题进行求解.

本文基于自适应大规模邻域搜索算法(adaptive large neighborhood search, ALNS)设计求解子问题的启发式规则,算法的具体描述如下:

1) 初始解

定义 N 为港口数量, M 为货源点数量,一个初始解可以表示为 $O, 1, 2, N+1, N+2, 6, N+3, O'$, 其中 O 和 O' 分别表示枢纽港和虚拟枢纽港, $1 \sim N$ 的数字分别表示喂给港的编号, $N+1 \sim N+M$ 的数字分别表示第 1 个到第 M 个货源点,即小于等于 N 的编号为喂给港;大于 N 的编号为货源点.

因此如上初始解所表示的船舶路径为 $O-1-2-6-O'$, 其中货源点 1 和货源点 2 被分配至喂给港 2, 货源点 3 被分配至喂给港 6.

2) 邻域搜索动作

针对初始解的生成规则 and 问题的特性,设计了四种邻域动作: (a) 随机插入-随机选择解的一个位置插入一个 $0 \sim M+N$ 之间的与原解不重复的数值. (b) 最优插入-评估所有插入位置以及所有可插入的不重复的数值,选择一个最好的插入位置和相应最好的插入数值对原解进行更新. (c) 随机移除-随机选择解的一个位置并移除该位置的数值. (d) 最优移除-评估所有可以移除的位置,选择一个最优的移除位置并移除该位置的数值.

在进行邻域动作时需要遵循以下规则:插入的数值必须与原解中的所有数值不重复;插入或移除的位置不能是解的最左边或者解的最右边(保证船舶必须从枢纽港出发并最终返回枢纽港).

3) 自适应邻域搜索规则

在算法最开始随机生成 10 个初始可行解,在进行邻域搜索时,随机在 10 个解中选择一个解进入邻域搜索操作.在进行邻域搜索之前,首先根据 4 种邻域动作的权重值(算法初始时,四种邻域动作的权重值相同,在迭代的过程中需要根据解的好坏进行更新),运用轮盘赌机制选择一个邻域动作,再对所选定的解进行该邻域动作下的邻域搜索.当每次通过邻域搜索得到新解比当前解更优时,则用新解替代当前解;若新解比当前解差,则以一定的概率接受新解成为当前解.设置能够使子问题目标函数值为负的解的集合为 S ,若新解的目标函数值小于 0,则将该解加入到集合 S 当中,并对该解进行记录,禁止加入重复的解到集合 S .

在选择邻域动作之前,还需要根据当前进行邻域搜索的解的长度对四种邻域动作当前的权重值进行临时的设置.邻域动作随机插入和最优插入的权重值依据式(41)进行临时设置;邻域动作随机移除和最优移除的权重值依据式(42)进行临时设置,即

$$\rho = (1 - \lambda)\rho + \lambda(M + N - L), \quad (41)$$

$$\rho = (1 - \lambda)\rho + \lambda L, \quad (42)$$

其中 λ 为 0-1 之间的小数, M 为货源点的数量, N 为港口的数量, L 表示当前要进行邻域搜索的解的长度.

根据式(41)和式(42)对邻域动作的权重进行临时的设置, 可以使得在原有权重的基础上, 长度越长的解越可能进行移除操作, 长度越短的解则越可能进行插入操作. 在选择完某一个邻域动作后, 还需要对四种邻域动作临时设置的权重值进行还原.

此外, 在完成一次邻域动作搜索后, 需要依据所得新解的好坏, 按照式(43)对当前所选邻域动作的权重值进行更新, 即

$$\rho = \mu \times \rho + (1 - \mu)\varphi, \quad (43)$$

其中 μ 为 0-1 之间的小数, φ 是当前进行的邻域搜索所得结果的评估值, 具体表示为

$$\varphi = \begin{cases} \omega_1, & \text{如果通过邻域搜索得到的新解是当前所得到的最优解} \\ \omega_2, & \text{如果通过邻域搜索得到的新解优于当前解} \\ \omega_3, & \text{如果通过邻域搜索得到的新解劣于或等同于当前解} \\ \omega_4, & \text{如果通过邻域搜索得到的新解是不可行解,} \end{cases} \quad (44)$$

其中 $\omega_1 > \omega_2 > \omega_3 > \omega_4 > 0$

4) 不同船型的处理方法

每次判断生成的新解是否可行时, 均按照最大船型的参数进行计算. 但在得到一个可行解后, 计算其成本以及 SP 模型目标函数之前, 需要判断是否可以在保证解可行的情况下, 更换更小的船型, 若可以的话则按照当前可更换的最小船型计算相应参数的值.

5) 算法停止规则

ALNS 算法迭代 10 000 次停止. 若集合 S 为非空, 则将 S 中所有解转化成为列加入到 RMP 模型中. 求解 RMP 模型, 得到新的对偶变量值后, 再进行 ALNS 算法; 若 S 为空则可看作当前 MP 模型中不存在检验数为负的列, 可以判断 MP 模型已经求得了最优解(近优解).

ALNS 算法的步骤如下所示:

步骤 1 随机生成 10 个初始可行解;

步骤 2 设置初始迭代次数 $\text{iter} \leftarrow 0$;

步骤 3 为所有的邻域动作设置相同的初始权重;

步骤 4 在 10 个解中随机选中一个解 s ;

步骤 5 按照解 s 的长度对四种邻域动作的权重值进行临时设置;

步骤 6 采用轮盘赌机制选择一个邻域动作, 对解 s 进行该邻域动作下的邻域搜索, 生成一个新解 s' , 并对临时设置的权重值进行还原. 若解 s' 优于解 s , 则用 s' 替换 s , 若解 s' 比解 s 差, 则以一定概率接受 s' ;

步骤 7 若解 s' 的目标函数值小于 0, 则将解 s' 加入至集合 S 中并对解 s' 进行记录, 防止其被重复加入集合 S ;

步骤 8 按照当前所选择的邻域动作的搜索结果, 对该邻域动作的权重进行更新;

步骤 9 $\text{iter} \leftarrow \text{iter} + 1$. 如果 $\text{iter} \leq 10\ 000$, 则转至步骤 4, 否则转至步骤 10;

步骤 10 输出集合 S , 算法终止.

4 实验结果与分析

本文基于珠江三角洲地区, 选取一个枢纽港、若干喂给港以及若干货源点构成三级支线运输网络. 设计不同规模的算例, 分别运用线性求解器 Gurobi、两阶段求解算法、整合优化求解算法进行求解, 以验证模型的准确性、算法的求解效率以及本文的研究意义. 实验的计算机参数配置为 Intel(R)Core(TM)I5-8265U, 1.80 GHz, 8 GB RAM. 算法在 Visual Studio 2017 C++ 中实现编码, 所使用的 Gurobi 版本为 8.1.0.

4.1 算例描述

- 1) 设置包括枢纽港在内的港口数量 $|P|$ 为 10, 15, 20, 25, 30; 货源点数量 $|C|$ 为 10 到 80 不等. 对于不同的 $(|P|, |C|)$ 组合, 共计生成 45 个算例.
- 2) 选取三种类型的支线集装箱船舶, 分别为 100 TEU, 150 TEU, 200 TEU.
- 3) 在距离枢纽港一定半径范围内随机生成指定数量的货源点, 各货源点的集装箱运输需求在 10 TEU~45 TEU 之间随机生成.
- 4) 各喂给港的卸箱量设置: 0.2 的概率为 0 TEU; 0.8 的概率在 15 TEU~55 TEU 之间随机生成.
- 5) 各港口的效率、时间窗限制以及各货源点集装箱最晚运至枢纽港的期限在一定范围内随机生成.

4.2 实验结果与对比分析

为了更好的展示三级支线运输网络中, 货源点集装箱分配对于支线船舶运输路径以及运输成本的影响, 分别使用两阶段求解算法和整合优化求解算法对 10 个货源点和 10 个港口的算例进行求解, 求解结果如表 3 所示.

表 3 两阶段求解算法vs.整合优化求解算法
Table 3 Two-phase algorithm vs. integrated optimization algorithm

	两阶段求解算法	整合优化求解算法
货源点集装箱分配方案	1-2, 2-9, 3-1, 4-3, 5-2	1-5, 2-9, 3-1, 4-9, 5-2
(货源点编号—港口编号)	6-4, 7-2, 8-1, 9-6, 10-6	6-9, 7-2, 8-1, 9-6, 10-6
支线船舶调度方案	$O-9-O'(100 \text{ TEU})$ $O-2-1-O'(100 \text{ TEU})$ $O-7-5-3-8-6-4-O'(150 \text{ TEU})$	$O-9-O'(100 \text{ TEU})$ $O-2-1-O'(100 \text{ TEU})$ $O-7-4-5-3-8-6-O'(100 \text{ TEU})$
集装箱分配成本/ 10^3 元	1.05	1.08
支线船舶运输成本/ 10^3 元	32.45	25.93
总成本/ 10^3 元	33.50	27.01

在两阶段算法的求解结果中, 货源点的集装箱实现了最优分配, 得到了最小的集装箱分配成本 1 050 元. 在最优的集装箱分配方案下, 需要使用 2 艘 100 TEU 的集装箱船舶和 1 艘 150 TEU 的集装箱船舶实现最优的船舶调度, 相应的支线船舶运输成本为 32 450 元, 最终的总运输成本为 35 500 元; 在整合优化求解的结果中, 货源点 1 被分配至喂给港 5, 货源点 4 和货源点 6 被分配至喂给港 9, 均没有进行最优的分配, 集装箱分配成本增加至 1 080 元. 然而在该集装箱分配方案下, 仅需要 3 艘 100 TEU 的集装箱船舶即可实现支线船舶的最优调度, 船舶运输成本大大降低. 整合优化算法求得的总运输成本比两阶段求解算法求得的总运输成本降低 19.37%, 初步验证了构建三级支线运输网络以及进行整合优化的意义.

为了进一步验证算法的有效性和本文的研究意义, 分别运用 3 种求解方法对 45 个不同规模的算例进行求解, 求解结果如表 4 所示. “*”表示 Gurobi 运行 1 h 没有得到最优解, 相应数值为 Gurobi 运行 1 h 得到的满意解; “—”表示 Gurobi 运行 1 h 没有得到可行解; Gap1 为整合优化算法求解结果与 Gurobi 计算结果的对比情况; Gap2 为整合优化算法求解结果与两阶段算法求解结果的对比情况.

表 4 中结果显示, 使用线性求解器 Gurobi 求解模型(1)时, 只有算例 1、算例 2、算例 3、算例 10 和算例 19 在 1 h 内求得了最优解, 其余算例均无法在 1 h 内求得最优解或可行解. 而对于所有的算例, 整合优化求解算法均能在 12 min 内求得最优解(近优解), 平均求解时间为 469.29 s.

此外, 对 Gurobi 和整合优化算法的求解结果进行对比分析可以发现, 对于 Gurobi 可以在 1 h 内求得最优解的算例, 整合优化算法的求解结果与 Gurobi 的求解结果相差最多为 1.78%, 最少为 0, 平均相差 0.81%; 对于其他算例, 整合优化算法所求的成本均小于 Gurobi 求解 1 h 所得到的成本.

图 4 为表 4 中 Gap1 值的柱状图, 进一步展示了除了 4 个算例之外, 整合优化算法的求解结果均优于或等于 Gurobi 的求解结果, 成本最多可降低 22.47%, 平均降低 7.44%. 因此, 通过对比 Gurobi 以及整合优化算法的求解结果, 验证了整合优化算法的求解效率.

表4 三种求解方法下不同规模算例的求解结果
Table. 4 Computational results of three types of methods for different size of instances

编号	(P , C)	Gurobi		两阶段算法		整合优化求解算法		算法对比结果	
		成本/ 10^3 元	时间/s	成本/ 10^3 元	时间/s	成本/ 10^3 元	时间/s	GAP1/%	GAP2/%
1	(10,10)	27.01	121.07	33.50	0.45	27.01	152.38	0.00	-19.37
2	(10,15)	27.74	717.68	33.97	0.53	28.03	243.68	1.05	-17.49
3	(10,20)	38.11	3 315.13	49.40	0.78	38.47	305.28	0.94	-22.13
4	(10,25)	40.08*	3 600.00	55.23	0.87	39.53	375.94	-1.37	-28.43
5	(10,30)	49.31*	3 600.00	58.76	0.88	47.30	498.26	-4.08	-19.50
6	(10,35)	59.40*	3 600.00	59.55	0.81	59.34	539.56	-0.10	-0.35
7	(10,40)	65.64*	3 600.00	68.93	0.92	63.55	559.28	-3.18	-7.81
8	(10,45)	76.01*	3 600.00	78.95	0.94	71.19	537.28	-6.34	-9.83
9	(10,50)	86.69*	3 600.00	94.25	1.28	80.85	567.26	-6.74	-14.22
10	(15,10)	33.21	3 304.12	40.44	0.85	33.80	149.21	1.78	-16.42
11	(15,15)	38.56*	3 600.00	47.36	1.09	38.22	179.33	-0.88	-19.30
12	(15,20)	41.98*	3 600.00	51.56	1.49	41.01	336.25	-2.31	-20.46
13	(15,25)	46.76*	3 600.00	59.26	1.47	44.80	422.23	-4.19	-24.40
14	(15,30)	55.86*	3 600.00	69.15	1.72	53.37	476.27	-4.46	-22.82
15	(15,35)	67.77*	3 600.00	75.63	1.46	59.98	559.43	-11.49	-20.69
16	(15,40)	72.35*	3 600.00	76.38	1.74	64.98	547.87	-10.19	-14.93
17	(15,45)	86.64*	3 600.00	87.63	2.64	73.73	588.72	-14.90	-15.86
18	(15,50)	—	3 600.00	98.52	2.23	82.03	575.25	—	-16.74
19	(20,10)	46.11	2 968.57	62.05	1.94	46.24	297.13	0.28	-25.48
20	(20,15)	50.39*	3 600.00	68.48	1.76	49.67	312.27	-1.43	-27.47
21	(20,20)	56.75*	3 600.00	73.12	1.93	53.28	336.14	-6.11	-27.13
22	(20,25)	60.96*	3 600.00	76.56	1.69	55.98	421.22	-8.17	-26.88
23	(20,30)	64.90*	3 600.00	78.17	2.12	63.22	454.32	-2.59	-19.12
24	(20,35)	75.64*	3 600.00	86.47	1.32	68.32	530.21	-9.68	-20.99
25	(20,40)	85.63*	3 600.00	111.41	1.78	75.44	567.32	-11.90	-32.29
26	(20,45)	92.90*	3 600.00	113.14	1.71	81.62	596.93	-12.14	-27.86
27	(20,50)	—	3 600.00	120.73	1.88	89.66	602.31	—	-25.74
28	(25,10)	72.07*	3 600.00	97.95	1.84	63.93	398.27	-11.29	-34.73
29	(25,15)	73.26*	3 600.00	104.58	1.88	66.20	416.24	-9.64	-36.70
30	(25,20)	75.35*	3 600.00	112.37	1.96	70.65	438.98	-6.24	-37.13
31	(25,25)	80.66*	3 600.00	124.65	1.78	72.31	487.53	-10.35	-41.99
32	(25,30)	89.06*	3 600.00	128.27	1.99	77.89	506.21	-12.54	-39.28
33	(25,35)	103.86*	3 600.00	133.22	1.67	84.25	556.77	-18.88	-36.76
34	(25,40)	102.37*	3 600.00	135.62	1.85	86.92	591.33	-15.09	-35.91
35	(25,45)	—	3 600.00	144.67	1.92	92.57	587.30	—	-36.01
36	(25,50)	—	3 600.00	149.66	1.87	101.20	621.88	—	-32.38
37	(30,10)	112.23*	3 600.00	137.42	1.94	95.22	402.77	-15.16	-30.71
38	(30,15)	131.61*	3 600.00	146.28	1.86	104.54	425.62	-20.57	-28.53
39	(30,20)	140.25*	3 600.00	156.29	1.90	108.73	447.32	-22.47	-30.43
40	(30,30)	—	3 600.00	171.57	2.18	116.25	490.30	—	-32.24
41	(30,40)	—	3 600.00	190.22	2.26	120.69	535.26	—	-36.55
42	(30,50)	—	3 600.00	205.76	2.15	125.81	593.87	—	-38.86
43	(30,60)	—	3 600.00	223.60	2.45	128.33	585.39	—	-42.61
44	(30,70)	—	3 600.00	237.73	2.71	136.56	629.92	—	-42.56
45	(30,80)	—	3 600.00	248.61	2.87	141.95	671.85	—	-42.90
平均值		69.35	3 431.70	106.16	1.67	73.88	469.29	-7.44	-26.67

注: GAP1 = (整合优化求解算法得到的成本 - Gurobi 得到的成本) / Gurobi 得到的成本

GAP2 = (整合优化求解算法得到的成本 - 两阶段算法得到的成本) / 两阶段算法得到的成本

此外, 对比观察表4中整合优化算法的求解结果与两阶段算法的求解结果可知, 对于所有的算例, 整合

优化算法的求解结果均要优于两阶段算法的求解结果. 在 45 个算例中, 既存在集装箱分配发生变化后可以将大船换成小船使总成本降低的情况, 也存在集装箱分配发生变化后, 集装箱船舶的行驶路线发生改变而使总成本降低的情况.

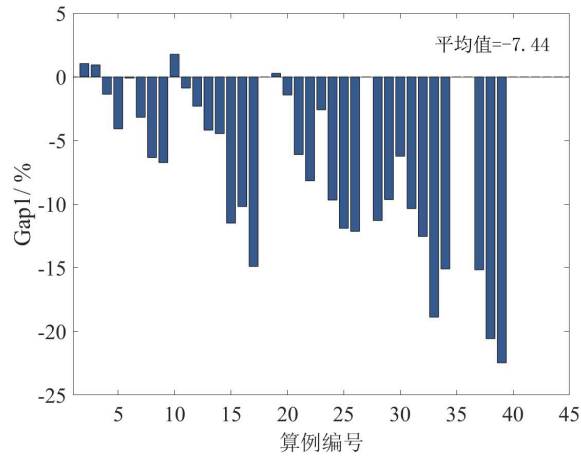


图 4 Gap1 柱状图

Fig. 4 The bargraph of Gap1

图 5 为表 4 中 Gap2 值的柱状图, 由图 5 可知, 整合优化集装箱分配与支线船舶调度最多可使成本降低 40 % 以上, 平均成本降低值为 26.67 %, 充分验证了整合优化求解的必要性以及本文的研究意义.

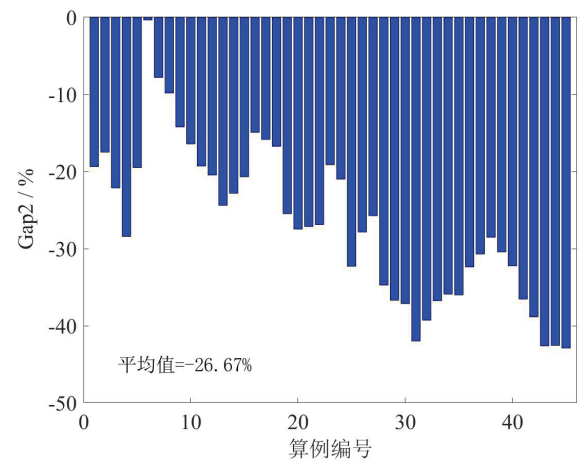


图 5 Gap2 柱状图

Fig. 5 The bargraph of Gap2

为证明使用多船型集装箱船舶对于本文所研究问题的优势, 选取表 4 中算例 28~36 共 9 个算例, 运用整合优化求解算法, 分别对多船型、仅 100 TEU 船型、仅 150 TEU 船型、仅 200 TEU 船型 4 种情况进行求解, 求解结果如图 6 所示.

由图 6 可知, 对于所有算例, 使用多船型的调度方案均比单一船型要好, 且差别较为明显, 因此验证了使用多船型集装箱船舶进行集装箱分配与支线船舶调度联合优化的必要性.

5 结束语

为顺应航运企业向供应链终端整合的发展趋势, 同时考虑到货源点集装箱分配方案对于支线集装箱船舶调度成本的影响, 本文将有集装箱运输需求的货源点加入传统的支线船舶运输网络进行研究, 构建了三

级支线运输网络. 以运输总成本最小化为目标, 建立了集装箱分配与集装箱船舶调度联合优化模型, 并分别采用两阶段求解算法以及基于列生成原理的整合优化算法对问题进行求解. 实验结果表明, 整合优化求解算法相对于线性求解器 Gurobi 更具优势, 由此验证了整合优化算法的有效性; 对比两阶段算法与整合优化算法的求解结果, 整合优化算法能够得到更低的支线网络运输总成本, 验证了整合优化的重要性.

本研究可为集装箱船舶运输企业向供应链终端延伸服务的愿望和计划提供建议, 促使其与产生集装箱运输需求的客户方进行合作, 提前预知运输需求, 获得能够使支线运输总成本最低的集装箱分配计划与船舶调度方案, 从而进一步加强航运企业同内陆腹地的连接. 将集装箱运至枢纽港进行后续的干线运输有明确的时间要求, 因此本文的研究主要侧重于三级支线网络的集运过程. 而对于网络末端的配送问题, 则具有更强的不确定性, 需要考虑喂给港的堆场容量、外集卡的提箱时间、堆场的堆存成本等诸多因素. 在未来的研究中, 拟在此研究基础上, 对三级支线网络疏运过程中的协同优化问题开展进一步讨论.

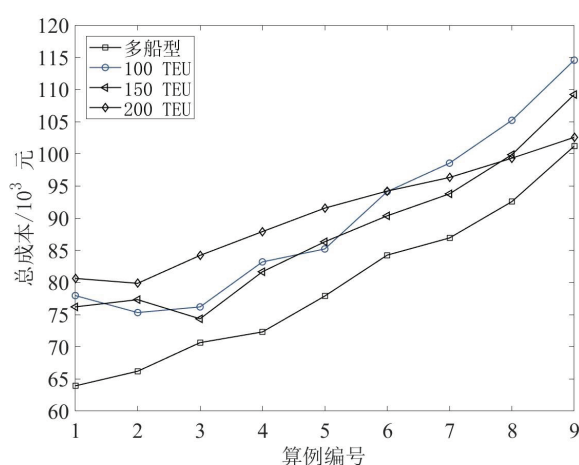


图6 多船型调度与单一船型调度成本对比

Fig. 6 The difference between the cost for the scheduling of single-type containership and multi-type containership

参考文献:

- [1] 赵宇哲, 周晶淼, 迟国泰. 竞争环境下轴-辐式海运网络设计与定价决策. 系统工程学报, 2018, 33(5): 615–626.
Zhao Y J, Zhou J M, Chi G T. Hub-and-spoke shipping network design and pricing decision in a competitive environment. Journal of Systems Engineering, 2018, 33(5): 615–626. (in Chinese)
- [2] Msakni M K, Fagerholt K, Meisel F, et al. Analyzing different designs of liner shipping feeder networks: a case study. Transportation Research, Part E: Logistics and Transportation Review, 2020, 134: 101839.
- [3] Zheng J F, Meng Q, Sun Z. Liner hub-and-spoke shipping network design. Transportation Research, Part E: Logistics and Transportation Review, 2015, 75: 32–48.
- [4] 赵刚. 班轮航线配船模型的分析与改进. 系统工程学报, 1997, 12(1): 82–88.
Zhao G. Analysis and improvement on the vessel allocation model of liner service. Journal of Systems Engineering, 1997, 12(1): 82–88. (in Chinese)
- [5] Gelareh S, Maculan N, Mahey P, et al. Hub-and-spoke network design and fleet deployment for string planning of liner shipping. Applied Mathematical Modelling, 2013, 37(5): 3307–3321.
- [6] Meng Q, Zhao H, Wang Y. Revenue management for container liner shipping services: critical review and future research directions. Transportation Research, Part E: Logistics & Transportation Review, 2019, 128: 280–292.
- [7] Christiansen M, Hellsten E, Pisinger D, et al. Liner shipping network design. European Journal of Operational Research, 2020, 286(1): 1–20.
- [8] Sambracos E, Paravantis J A, Tarantilis C D, et al. Dispatching of small containers via coastal freight lines: The case of the Aegean sea. European Journal of Operational Research, 2004, 152(2), 365–381.
- [9] 靳志宏, 解玉真, 李阳, 等. 集装箱支线运输船舶调度优化问题. 中国航海, 2008, 31(4): 415–419.
Jin Z H, Xie Y Z, Li Y, et al. Scheduling optimization problems of feeder liner container ships. Navigation of China, 2008, 31(4): 415–419. (in Chinese)

- [10] Karlaftis M G, Kepaptsoglou K, Sambracos E. Containership routing with time deadlines and simultaneous deliveries and pick-ups. *Transportation Research, Part E: Logistics and Transportation Review*, 2009, 45(1): 210–221.
- [11] Ji M J, Shen L X, Shi B S, et al. Routing optimization for multi-type containerships in a hub-and-spoke network. *Journal of Traffic and Transportation Engineering (English Edition)*, 2015, 2(5): 362–372.
- [12] Santini A, Plum C E M, Ropke S. A branch-and-price approach to the feeder network design problem. *European Journal of Operational Research*, 2018, 264(2): 607–622.
- [13] 计明军, 张开放, 祝慧灵, 等. 基于可变航速的支线集装箱船舶调度优化模型与算法. *运筹与管理*, 2019, 28(11): 18–26.
Ji M J, Zhang K F, Zhu H L, et al. Optimization model and algorithm of feeder line containership scheduling based on variable speed. *Operational Research and Management Science*, 2019, 28(11): 18–26. (in Chinese)
- [14] Ghoniem A, Scherrer C R, Solak S. A specialized column generation approach for a vehicle routing problem with demand allocation. *Journal of the Operational Research Society*, 2013, 64(1): 114–124.
- [15] Solak S, Scherrer C, Ghoniem A. The stop-and-drop problem in nonprofit food distribution networks. *Annals of Operations Research*, 2014, 221: 407–426.
- [16] Reihaneh M, Ghoniem A. A branch-and-price algorithm for a vehicle routing with demand allocation problem. *European Journal of Operational Research*, 2019, 272(2): 523–538.
- [17] Dantzig G B, Wolfe P. Decomposition principle for linear programs. *Operations Research*, 1960, 8(1): 101–111.
- [18] Victoria J F, Afsar H M, Prins C. Column generation based heuristic for the vehicle routing problem with time-dependent demand. *IFAC-Papers On Line*, 2016, 49(12): 526–531.
- [19] Thomas C, Cortés Cristián E, Michel G, et al. A column generation approach for location-routing problems with pickup and delivery. *European Journal of Operational Research*, 2019, 272(1): 121–131.
- [20] Feillet D, Dejax P, Gendreau M, et al. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 2004, 44(3): 216–229.

作者简介:

孔灵睿(1995—), 女, 内蒙古呼伦贝尔人, 博士, 研究方向: 交通运输规划, 物流系统优化, Email: klr@dlmu.edu.cn;

计明军(1973—), 男, 内蒙古包头人, 博士, 研究方向: 物流系统优化与模拟仿真, 交通运输规划, Email: jmj@dlmu.edu.cn;

孙以宁(1995—), 男, 吉林吉林人, 硕士, 研究方向: 物流系统优化, Email: 18041105797@163.com;

关云潇(1995—), 男, 辽宁铁岭人, 硕士, 研究方向: 物流系统优化, Email: 18018980610@163.com;

郭兴海(1989—), 男, 黑龙江哈尔滨人, 博士, 研究方向: 物流控制与管理, Email: xocan0612@163.com.