doi:10.3772/j.issn.1002-0470.2025.03.009

多舱位多行程带装载组合约束的成品油二次配送车辆路径优化①

王 成②*** 余浩男* 郭恒均* 张文珠③*

(*浙江工业大学机械工程学院 杭州 310023)

(**浙江工业大学台州研究院 台州 318001)

摘 要 针对多舱位多行程带装载组合约束的成品油二次配送车辆路径问题进行研究。已知油库使用多种车型的多舱位油罐车对加油站多种类型的成品油订单进行配送,每个油品订单只能装载到一个车舱,多个订单在同一车辆上的装载需要满足一定的装载组合约束。考虑车辆的配送成本和未满足订单的损失成本,以综合配送成本最小为目标建立问题的混合整数规划模型。为求解该模型,根据先分组后路径的求解策略,设计了一种变邻域禁忌搜索启发式算法,该算法在改进最近邻插入算法构造的初始解的基础上,应用"移除-重插入"的邻域操作进行优化从而获得近似最优解。算例实验结果表明,变邻域禁忌搜索算法可以在短时间内对问题进行有效的求解。所提出的模型和算法可以为成品油销售企业优化成品油二次配送提供理论依据和决策支持。

关键词 成品油二次配送;多舱位;多行程;装载组合约束;路径优化

成品油二次配送(以下简称成品油配送)作为油品供应链的最终环节,其目的是将成品油通过油罐车从油库配送至各个加油站,从而满足加油站的补货需求。在成品油物流体系中,公路配送成本占总物流成本的比重高达 60% ~70%[1],优化潜力巨大,因此越来越多的石油企业将优化成品油配送路径作为降低运营成本、增加物流效益的重要突破口。

成品油配送问题本质上是一个车辆路径问题(vehicle routing problem,VRP),除了具备一般 VRP 多产品、多车型的特征外,还带有油品运输场景的特殊性和复杂性。根据不同配送场景和优化策略,国内外学者对成品油配送问题进行了广泛的研究与讨论。文献[2-5]分别针对多周期、多车型、多油库的成品油多舱配送问题进行研究,并取得了较好的成果。文献[6]研究了多行程带补货时间窗的成品油多舱配送问题,设计了包含内外两层循环的启发式算法。文献[7]构建了多行程多交货期的成品油配送模型,设计了一种改进的变邻域搜索算法(variable neighborhood search,VNS)。文献[8]研究了一个多舱位、多行程、需求可拆分的油品补货问题,并应用自适应大邻域搜索算法进行求解。文献[9]研究了多车舱、多趟次、车舱容量可拆分的成品油路径规划问题。文献[10]讨论了主动配送模式下的加油站补货问题,并构建了相应混合整数规划模型。文献[11]针对多油库、多车型、多行程、带隔舱和软时间窗的成品油配送问题,应用了一种混合蚁群算法进行求解。文献[12,13]考虑了成品油多舱配送中油品装卸顺序和工作量均衡等约束的影响,分别设计了高效求解的启发式算法。综上所述,目前成品油配送的相关研究主要集中在多车型、多舱位、多行程等方面,但随着成品油消费规模不断扩大,物流配送服务日趋精细化,这对成品油配送提出了更高要求。为保证油品运输的安全,不同种类的油品在同一油罐车不同舱位上装载时要符合严格的规范,

① 国家社科基金项目(20FJYB020)和教育部人文社科研究项目(19YJCZH020)资助。

② 男,1982 年生,博士,副教授;研究方向:复杂系统建模与优化,智能优化算法;E-mail:cwang@zjut.edu.cn。

③ 通信作者,E-mail:wzzhang@zjut.edu.cn。

(收稿日期:2023-05-08)

即油品的装载组合要满足一定的排列顺序。在考虑这种装载组合约束的情况下,优化油罐车配送路径的同时还必须保证油品对应车舱装载的可行性,因此比一般的成品油配送车辆路径问题具有更大的求解难度,而现有文献尚未有解决这类问题的有效算法。

本文拟对带装载组合约束的多产品、多车型、多舱位、多行程的成品油配送车辆路径优化问题(multi-compartment multi-trips vehicle routing problem with loading combination constraints, MCMTVRPLC)开展研究。以综合配送成本最小为目标,建立混合整数规划模型,基于先分组后路径的求解策略,设计一种变邻域禁忌搜索算法(tabu search, TS)对模型进行求解,并通过算例实验验证模型和算法的有效性。

1　问题描述与模型构建

1.1　问题描述

多舱位多行程带装载组合约束的成品油二次配送车辆路径问题可以被描述为:针对某地区由一个油库和多个加油站所构成的成品油配送网络,调度中心根据加油站提交的油品补货订单,对多舱油罐车进行路径规划和行程指派,将油品从油库配送至加油站。已知各加油站的油品订单以及油库可使用的油罐车的型号及数量,每个油品订单只能由一辆油罐车一次配送满足,订单对应的油品只能装载到一个隔舱中,不同订单不能混装,多个油品订单由同一油罐车配送时需要满足装载组合约束。制定成品油配送方案,从而在尽可能满足加油站油品需求的前提下最大化配送效益。

由于成品油实际配送中油罐车数量有限,往往会遇到油品需求无法配送或无法完全配送的情况。本文考虑了这2种情况,为了简化计算,对所有未配送的油品需求量求和,将其按容量折算为一个惩罚成本,用以表示委托配送这部分油品的外包费用,与油罐车固定使用成本、油罐车运输成本共同构成成品油配送的综合配送成本。

1.2　模型假设

为简化模型,本文做出如下假设。

(1) 油库拥有充足的供应量,能够满足所有加油站的油品需求,所有油罐车均在油库完成油品的装载后发车,完成配送任务后返回油库。

(2) 加油站可以提出多个油品订单,单个订单只能由一辆油罐车配送,不同订单可以由不同油罐车配送。任意订单的油品需求量均不超过可用车型中最大车舱的装载量。

(3) 油罐车在同一行程中不允许重复访问相同加油站,即不存在访问回路。

(4) 成品油属于液体危险品,为避免运输过程中油品晃动影响车辆的平稳行驶,油品装载需要满足车舱配载约束。

(5) 为了提高车辆利用率,在保证订单一定满足率的情况下,允许订单部分不被配送,从而匹配车舱的装载容量。

(6) 多个油品订单由同一油罐车配送时,需要满足订单的油品种类编号越小,其对应装载的车舱编号也越小的约束。

1.3　符号和变量说明

定义以下符号。

(1)集合

$N = \{0,1,\cdots,|N|\}$ 为节点集合;其中$\{0\}$表示油库,即路线的起点和终点。

$N' = \{1,\cdots,|N|\}$ 表示加油站集合。

$K = \{1,2,\cdots,|N|\}$ 为可用油罐车集合。

$P_i = \{1,2,\cdots,|P_i|\}$ 为加油站的油品订单集合。

$M_k = \{1,2,\cdots,|M_k|\}$ 为油罐车 k 的车舱集合。

$R_k = \{1,2,\cdots,|R_k|\}$ 为油罐车 k 的行程集合。

(2)下标

i、j 是加油站和油库编号, i、$j \in N$。

k 为油罐车编号, $k \in K$。

p 为订单编号, $p \in P_i$。

m 为车舱编号, $m \in M_k$。

r 为行程编号, $r \in R_k$。

(3)参数

t_{ij} 表示节点 i 到节点 j 的行驶时间。

d_{ij} 表示节点 i 到节点 j 的路线距离。

q_{ip} 表示加油站 i 的订单 p 的需求量。

o_{ip} 表示加油站 i 的订单 p 的油品种类。

v_{km} 表示油罐车 k 的车舱 m 的容量。

s 表示油品单位容量装载和卸载速率。

$T_{\max}$ 表示油罐车最大工作时间。

α 表示车舱最低配载率。

β 表示订单最低满足率。

c_1 表示车辆固定使用成本。

c_2 表示车辆单位距离运输成本。

c_3 表示油品未配送需求单位惩罚成本。

λ 表示一个很大的正数。

(4)决策变量

u_{kr}: 油罐车 k 在行程 r 进行了配送,则 $u_{kr}=1$;否则 $u_{kr}=0$。

x_{ijkr}: 如果油罐车 k 在行程 r 从节点 i 行驶到节点 j, 则 $x_{ijkr}=1$;否则 $x_{ijkr}=0$。

y_{ikpmr}: 如果加油站 i 的订单 p 在油罐车 k 的行程 r 中装载到车舱 m 上,则 $y_{ikpmr}=1$;否则 $y_{ikpmr}=0$。

1.4 模型建立

建立 MCMTVRPLC 的整数规划模型如下。

$$\min f = c_1\sum_{k\in K}u_{k1} + c_2\sum_{i,j\in N,i\neq j}\sum_{k\in K}\sum_{r\in R_k}d_{ij}x_{ijkr} + c_3\sum_{k\in K}\sum_{r\in R_k}\sum_{m\in M_k}\sum_{j\in N'}\sum_{p\in P_j}[(1-y_{jkpmr})q_{jp} + y_{jkpmr}\max(0,q_{jp}-v_{\mathrm{km}})] \quad (1)$$

$$\text{s.t.}\sum_{j\in N'}x_{0jkr} = \sum_{i\in N'}x_{i0kr} = u_{kr},\ \forall k\in K,\forall r\in R_k \quad (2)$$

$$u_{k(r+1)}\leqslant u_{kr},\ \forall k\in K,\ r\in R_k\backslash\{|R_k|\} \quad (3)$$

$$\sum_{i\in N,i\neq j}x_{ijkr} = \sum_{i\in N,i\neq j}x_{jikr},\forall j\in N',\forall k\in K,\forall r\in R_k \quad (4)$$

$$\sum_{i,j\in S,i\neq j}x_{ijkr}\leqslant |S|-1,\forall S\in N',|S|\geqslant 2,\ \forall k\in K,\forall r\in R_k \quad (5)$$

$$\sum_{j\in N'}\sum_{p\in P_j}y_{jkpmr}\leqslant u_{kr},\forall k\in K,\forall r\in R_k,\forall m\in M_k \quad (6)$$

$$\sum_{k\in K}\sum_{m\in M_k}\sum_{r\in R_k}y_{jkpmr}\leqslant 1,\forall j\in N',\forall p\in P_j \quad (7)$$

$$\alpha\sum_{m\in M_k}y_{jkpmr}v_{km}\leqslant q_{jp}\sum_{i\in N,i\neq j}x_{ijkr},\forall j\in N',\ \forall k\in K,\forall p\in P_j,\forall r\in R_k \quad (8)$$

$$\beta\sum_{p\in P_j}y_{jkpmr}q_{jp}\leqslant v_{km}\sum_{i\in N,i\neq j}x_{ijkr},\forall j\in N',\ \forall k\in K,\forall m\in M_k,\forall r\in R_k \quad (9)$$

$$o_{jp}\sum_{j\in N'}\sum_{p\in P_j}(y_{jkpnr}-y_{jpkmr})\leqslant\lambda(1-\sum_{j\in N'}\sum_{p\in P_j}y_{jpkmr}),\ \forall k\in K,\forall m,n\in M_k,m>n,\forall r\in R_k \quad (10)$$

$$T_{\max}\geqslant 2s\sum_{r\in R_k}\sum_{m\in M_k}\sum_{j\in N'}\sum_{p\in P_j}y_{jpkmr}\min(q_{jp},v_{km}) + \sum_{r\in R_k}\sum_{i\in N}\sum_{j\in N,j\neq i}x_{ijkr}t_{ij},\forall k\in K \quad (11)$$

$$u_{kr}\in\{0,1\},\forall k\in K,\forall r\in R_k \quad (12)$$

$$x_{ijkr}\in\{0,1\},\forall i,j\in N,i\neq j,\forall k\in K,\forall r\in R_k \quad (13)$$

$$y_{ikpmr}\in\{0,1\},\forall i\in N',\forall k\in K,\forall p\in P_i,\ \forall m\in M_k,\forall r\in R_k \quad (14)$$

上述模型中,式(1)表示最小化综合成本函数,其中第 1 项为车辆的固定使用成本,第 2 项为车辆的可变运输成本,第 3 项为订单未满足的需求量的惩罚成本。式(2)表示车辆每次行程都从油库出发,最终返回油库。式(3)表示只有前一行程完成后,才能开始下一行程的配送。式(4)表示流入流出平衡约束。式(5)为子回路消除约束。式(6)表示车辆在配送时每个车舱有且仅能装载一个油品订单。式(7)表示对于每个订单最多只能满足一辆车辆的一个车舱一次配送。式(8)表示订单如果装载到车舱上,车舱的装载率不能低于最低配载率 α。式(9)表示允许订单部分油品不被配送,但订单的实际送达率不能低于最低满足率 β。式(10)为装载组合约束,表示多个油品订单由同一油罐车一次配送时,订单的油品种类编号越小,其对应装载的车舱编号也越小。式(11)表示车辆所有行程的运输和服务时间之和不能超过最大工作时间。式(12)~(14)表示决策变量约束。

2 变邻域禁忌搜索算法设计

由于 MCMTVRPLC 问题除具备一般 VRP 的复杂度外,还具备多舱位、多行程、带装载组合约束等特点,属于非确定性多项式难问题(non-deterministic polynomial-hard, NP)[14],所以如何设计快速高效的算法是解决此类问题的关键。禁忌搜索算法和变邻域搜索算法都是关于局部搜索的启发式算法,在求解车辆路径[15-17]、车间调度[18-19]等组合优化问题方面都有重要的应用。结合变邻域搜索算法扩大邻域

搜索范围和禁忌搜索算法跳出局部最优、避免迂回搜索的优势,本文将基于先分组后路径的思想,设计一种变邻域禁忌搜索的启发式算法(variable neighborhood tabu search heuristic algorithm,VNTS)对问题模型进行求解。

2.1　先分组后路径的求解策略

成品油配送问题需要根据油品的需求量以及车舱的容量从而确定油品在车舱上的指派,属于带有能力约束的车辆路径问题(capacitated vehicle routing problem,CVRP)。针对 CVRP 的启发式算法设计,主要有先分组后路径(cluster-first route-second)和先路径后分组(route-first cluster-second)2 种求解策略[20],如图 1 所示。先分组后路径的思想为先确定车辆要配送的站点集合,再优化车辆访问站点的路线;先路径后分组的思想则是先获取包含所有站点的大回路,再切割大回路生成配送路径,进而将其指派给车辆。

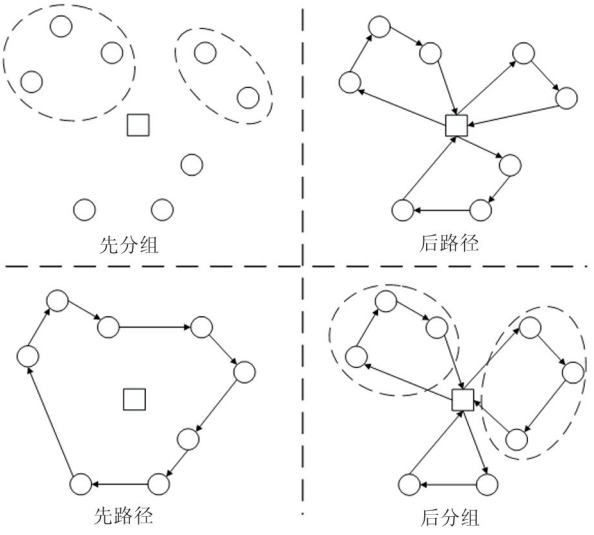

图 1　针对 CVRP 2 种求解策略的示例

CVRP 的求解可以被划分为路径和装载 2 个部分,其中路径部分用于降低车辆运输成本,装载部分用于检查给定路径是否存在可行的装载方案。对于本文研究的问题,由于存在装载组合约束以及车舱配载的限制,使得保证油品订单对应车舱装载的可行性成为了问题求解的主要难点。因此,本文采取先分组后路径的求解策略,先获取可行的订单装载方案,再考虑车辆配送路径的优化,从而将算法求解分为订单分组和路径规划 2 个过程。先划分车辆可配送的订单组合,完成订单对应车舱的指派;再确定车辆访问站点的顺序;最后生成配送路径。

2.2　解的编码形式

解的编码是算法设计的重要内容,对算法的求解性能有很大的影响。对于成品油配送问题,最终输出的配送方案需要包含订单对应车舱的指派(装载方案)和车辆访问站点的顺序(路径方案)两方面的内容。对于任意车辆任意行程的配送方案,采用“订单编号 + 车舱编号”的 2 元组定义节点元素 $Node(i,m)$,表示油品订单 i 与装载车舱 m 的配对关系。定义有序集合 $Y = \{Node_1,\cdots,Node_n\}$,其中节点元素的排列顺序表示车辆配送订单的访问顺序。例如给定集合 $Y = \{(1,2),(3,1),(2,3)\}$,表示车辆先配送装载在车舱 1 中的订单 2,再配送装载在车舱 3 中的订单 1,最后配送装载在车舱 2 中的订单 3,由于订单对应站点的信息已知,进而可获得车辆访问站点的路径。

由模型可知,MCMTVRPLC 问题的解可以看作是一个 3 层的结构,由上到下分别为总方案 S、车辆方案 V_k 和行程方案 T_r。行程配送方案 T_r 采取上述定义的 2 元组有序集合的形式存储配送信息,车辆配送方案 $V_k = \{T_1,\cdots,T_r,\cdots,T_m\}$,总体配送方案 $S = \{V_1,\cdots,V_k,\cdots,V_n\}$。

2.3　VNTS 整体框架

对于成品油配送这类复杂组合优化问题,如何平衡求解算法的探索(exploration)和利用(exploitation)2 种能力之间的关系是启发式算法设计的核心目标之一。本文基于“移除-重插入”的机制定义多种邻域结构对解的部分订单与车舱的匹配关系进行破坏和重构,在避免迭代中非可行解产生的同时,实现对邻域空间较大规模的探索。结合先分组后路径的思路以及解的编码,本文以解的一次修改的邻域操作为禁忌对象进行禁忌表的设置,从而减少迭代过程中的重复操作。此外,本文还应用集中性和多样性策略分别加强对优良解邻域和未知邻域的探索,以提高算法的求解效率。

为了利用算法求解过程中产生的历史方案信息,本文引入“方案池”的概念,用来存储迭代中生

成的不同订单组合对应车辆指派的优质配送信息，并将其应用于配送方案的生成和再优化。

定义 VNTS 中使用的符号和变量如下。

$iter$ 表示当前迭代次数，$iter_{max}$ 表示最大迭代次数，b 表示当前最优解连续保持不变的次数，b_{max} 表示当前最优解连续保持不变的最大迭代次数，c 表示当前候选集中候选解的数量，CL 表示候选集长度，DL 表示候选集中集中性元素和多样性元素的分界点长度。

VNTS 的整体框架如下所述。

步骤 1　初始化方案池、禁忌表、候选集和其他算法参数。

步骤 2　利用改进的最近邻启发式算法生成初始可行解，将其设置为当前解 X 和当前最优解 S_{best}，令 $iter=1$，$b=0$，$c=0$。

步骤 3　若 $c<DL$，则基于集中性策略选择邻域动作；若 $DL\leqslant c<CL$，则基于多样性策略选择邻域动作；对当前解 X 执行选定的邻域变换操作，生成新的可行解，结合方案池对解进行再优化，将其保存到候选集合中，令 $c=c+1$。

步骤 4　若 $c<CL$，则转步骤 3；否则，转步骤 5。

步骤 5　基于藐视准则判断当前候选解集中是否存在优于 S_{best} 的禁忌解。若存在，则解禁该解将其作为最佳候选解 X_{best}；否则，从非禁忌解中选择最优解作为最佳候选解 X_{best}。

步骤 6　如果 X_{best} 优于 X，则 $DL=DL+1$，否则 $DL=DL-1$。

步骤 7　令 $X=X_{best}$，$c=0$，$iter=iter+1$，更新禁忌表和方案池。若 X_{best} 优于 S_{best}，则令 $S_{best}=X_{best}$，$b=0$；否则，令 $b=b+1$。

步骤 8　若 $iter=iter_{max}$ 或 $b=b_{max}$，则算法停止迭代，输出目前最优解；否则，转到步骤 3。

VNTS 算法的流程图如图 2 所示。

2.4　VNTS 中各种操作的详细说明

2.4.1　基于改进的最近邻插入算法的初始解生成

本文在文献[21]提出的求解车辆路径问题的最近邻启发式的基础上，基于先分组后路径的求解策略，提出一种改进的最近邻插入算法（improved nearest-neighbor heuristic，INNH）用以求解成品油配送问题的初始可行解。其基本思想如下：首先，选择配送优先度最高的待配送加油站作为种子站点，生成初始订单配送组合，进而获取各油罐车对应组合的可行配送方案；其次，对每个方案进行检验，当存在车舱空闲时，从剩余的加油站中选择距离最近的站点，将其未配送的订单插入到当前订单组合中，并更新配送方案；然后，对所有可行的配送方案进行配送效益的评价，选择配送效益最大的方案作为最佳配送方案输出；最后，重复上述过程，直到所有订单被分配。

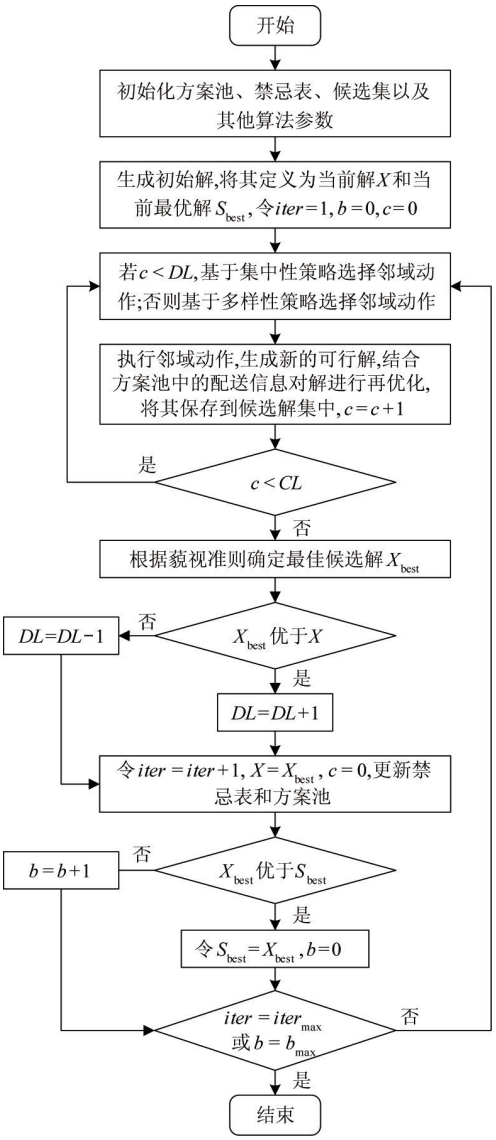

图 2　VNTS 流程图

关于种子站点和最佳配送方案的选择的详细步骤如下所述。

（1）种子站点的选择。令 U 为待配送加油站的

集合,K 为可使用油罐车的集合,P_i 为加油站 $i \in U$ 的待配送订单集合。计算每个加油站的配送优先度,按照公式 $i^* = \arg\max_{i \in U} Pri(i)$ 选择优先度最高的站点作为种子站点。加油站 i 的配送优先度 $Pri(i)$ 计算公式为

$$Pri(i) = \omega_1 D(i) + \omega_2 \sum_{p \in P_i} E(i, p) \quad (16)$$

$$E(i, p) = 1/(1 + \omega_3 \sum_{p \in P_i} Res(i, p, K) \quad (17)$$

式中:ω_1、ω_2、ω_3 为权重系数;$D(i)$ 表示加油站到油库的距离;$E(i, p)$ 表示站点的配送紧急度,配送紧急度越大则说明配送该站点的订单所能使用的车辆资源越少;$Res(i, p, K)$ 表示对于站点 i 的订单 p 可配送的车辆资源的数量。

(2)最佳配送方案的选择。令 S 为所有可行配送方案的集合,按照公式 $s^* = \arg\max_{s \in S} F(s)$ 选择配送效益最高的方案作为最佳配送方案,配送方案 s 的配送效益 $F(s)$ 计算公式为

$$F(s) = \gamma_1 f(s) + \gamma_2 ut(s) + \gamma_3 ur(s) \quad (18)$$

式中:γ_1、γ_2、γ_3 为权重系数;$f(s)$ 表示方案的配送成本;$ut(s)$ 表示方案车辆利用率,其大小为车辆实际运载量与额定运载量的比值;$ur(s)$ 表示方案所配送订单的紧急度之和。

改进的最近邻插入算法生成初始解的步骤如下所述。

步骤 1　初始化待配送加油站集合 U、可配送油罐车集合 K 以及车辆配送方案集合 S。

步骤 2　如果 $U \neq \varnothing$ 且 $K \neq \varnothing$,则转到步骤 3;否则,终止算法并输出 S。

步骤 3　创建配送方案的候选集合 D。计算 U 中每个站点的配送优先度,选择优先度最高的站点作为种子站点,生成初始订单配送组合,进而获得车辆相应可行的临时配送方案。

步骤 4　对所有的临时方案进行检验,判断方案中是否存在车舱空闲,如存在则按照总配送距离增量最小的原则从剩余待配送加油站中选择站点,将其未配送订单插入到当前订单组合中,生成新的配送方案,继续进行检验;否则,将当前配送方案保存到候选集合 D 中。

步骤 5　对候选集合 D 存储的配送方案进行配送效益的评价,选择配送效益最大的方案作为最佳配送方案,将其保存到集合 S 中,基于配送信息更新集合 U 和 K,转至步骤 2。

2.4.2　基于“移除-重插入”的邻域操作

对于本文研究的问题,传统的邻域动作(如交换、交叉、易位等)容易生成非可行解,影响算法求解效率。本文以订单与车舱的配对关系为操作对象,先采用移除操作从配送方案中选择订单移除,破坏解的部分结构;然后基于重插入操作将移除的订单重新插入,对解的结构进行修复,从而获得原可行解的邻域解。通过不同移动操作和重插入操作的组合,可以生成多种邻域算子,进行不同的领域动作。

本文采取的移除操作(removal)如下所述。

(1)随机移除。选择可行解的任意一条行程配送方案,从方案中随机移除一个存储订单与车舱的配对信息的节点元素 $Node(i,m)$,设置订单 i 为未配送订单,车舱 m 为空闲车舱。

(2)相似元素移除。若可行解存在未配送的订单 i,选择与订单 i 最相似的订单 j,将所在的节点 $Node(j,n)$ 从当前方案中移除;否则,先随机选择一个节点 $Node(i,m)$ 移除,再寻找与订单 i 最相似的订单 j 所在的节点 $Node(j,n)$ 移除。用式(19)来确定最相似的订单。

$$j^* = \arg\min_{j \in P}\left\{\lambda_1 d_{ij} + \lambda_2 \frac{|V_i - V_j|}{V_i} + \lambda_3\left(1 - \frac{|K_i \cap K_j|}{|K_i|}\right)\right\} \quad (19)$$

式中:λ_1、λ_2、λ_3 为权重系数,P 为已分配的订单集合,d_{ij} 表示订单 i、j 对应的加油站之间的距离,V_i 表示订单 i 的油品需求量,K_i 表示能够配送订单 i 的车辆资源集合。

(3)最大费用移除。计算配送方案中各节点的移除成本,选择移除成本最大的订单 i 所在节点 $Node(i,m)$ 移除。定义当前方案 s 的配送成本为 $f(s)$,移除订单 i 后,配送成本的变化值 $\delta(i,s) = f(s) - f_{-i}(s)$,则基于公式 $i^* = \arg\max_{i \in P} \delta(i,s)$ 确定移除成本最大的订单。

本文采取的重插入操作(reinsertion)描述如下。

(1)贪婪插入。基于贪婪准则,定义插入成本

$\Delta f_{i,x_{iN}}$ 为待配送订单 i 插入为新节点 N 的插入方案 x_{iN} 所引起的成本增加量,选择插入成本最低的插入方案,更新解的结构。

(2)后悔值插入。考虑一种前向的信息,令 x_{ik} 表示待配送订单 i 插入成本第 k 低的插入方案。定义后悔值 $R(i) = \Delta f_{i,x_{i1}} - \Delta f_{i,x_{i2}}$,表示订单 i 最优插入方案和次优插入方案对应插入成本的差值。选择后悔值最大的订单的最优插入方案,更新解的结构。

(3)适应值插入。结合式(18),定义适应值 $\Delta f_{i,x_{iN}}$ 为待配送订单 i 插入为新节点 N 的插入方案 x_{iN} 所引起的配送效益的增量,选择适应值最大的插入方案,更新解的结构。

2.4.3　集中性与多样性策略

在禁忌搜索算法中,集中性(intensification)搜索策略用于加强对优良解邻域的进一步探索,多样性(diversification)搜索策略用于加强对未知邻域的探索[22]。基于上述思想,本文将候选解的集合分为 2 个部分,这 2 个部分中的元素分别由基于集中性策略和多样性策略的邻域算子生成。对于集中性元素,希望对当前解的邻域进行深入探索,因此可以结合方案池中存储的优质配送信息对订单的移除和插入进行指导,进行有方向的探索。对于多样性元素,要尽可能探索未搜索过的区域,因此可以在选择订单移除和插入时考虑随机性。

2.4.4　禁忌规则设置

本文以移除和插入操作产生的邻域移动为禁忌对象设置禁忌列表,根据问题规模设置禁忌长度 L,从而达到跳出局部最优的目的。本文采用藐视准则对禁忌解中的某些优良状态进行特赦,尽可能地不错过产生最优解的移动。同时为避免算法陷入重复且无效的迭代,在迭代一定次数后,若最优解没有获得改进,则设置当前最优解为初始解,重新开始迭代并重置禁忌表。

3　算例分析

3.1　算例描述

基于实地调研,获取嘉兴某地区成品油二次物流配送的特征信息。已知该地区存在 1 个油库和 10 个加油站,油库及加油站坐标信息已知,如图 3 所示。油库下属一支多车型的运输车队,车辆型号及车舱的具体参数如表 1 所示。加油站的日常补货需求主要涉及 3 种汽油油品(92#、95#、98#),订单数量为 1 ~ 4 个,成品油配送相关参数设置如表 2 所示。为方便描述,设置油库编号为 0、加油站编号为 1 ~ 10,油品的种类编号为 1 ~ 3。某日加油站的补货订单如表 3 所示,油品订单由一个元组表示,例如订单(1,14 000,3)表示编号为 1 的订单,其对 3 号油品的需求量为 14 000 L。试寻找车辆最佳配送方案,使得综合配送成本最低。

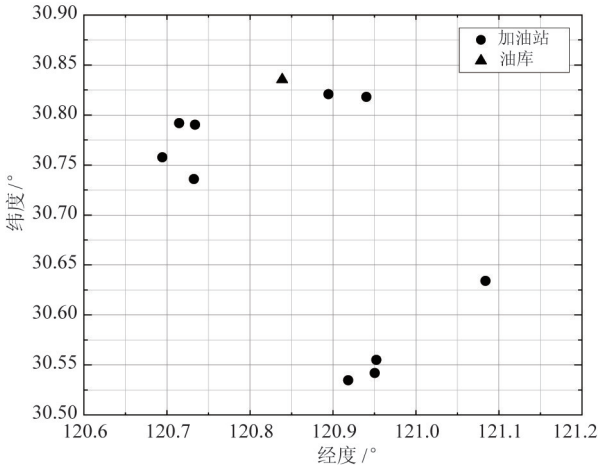

图 3　加油站及油库分布

表 1　油罐车相关参数

车辆编号	车舱数量	运载量/L			
		舱位 1	舱位 2	舱位 3	舱位 4
1	1	19 300	-	-	-
2	2	12 000	12 000	-	-
3	3	16 000	14 300	14 300	-
4	3	15 000	16 000	16 000	-
5	3	10 800	13 000	16 000	-
6	4	13 000	10 000	10 000	14 000

表 2　成品油配送相关参数设置

项目	值
车辆固定使用成本/(元·辆$^{-1}$)	200
单位距离运输成本/(元·km^{-1})	2.5
单位容量惩罚成本/(元·kL^{-1})	100
车辆最大工作时间/h	12
油品装卸速率/(kL·min^{-1})	1
车舱最低配载率	0.8
订单最低满足率	0.9

表 3　加油站补货订单信息

站点	油品订单
0	-
1	(1,14 000,3), (2,10 000,3)
2	(3,16 000,2), (4,16 000,3), (5,15 000,3)
3	(6,14 000,2), (7,10 000,2), (8,12 000,3)
4	(9,13 000,3), (10,16 000,3), (11,14 000,2)
5	(12,13 000,2), (13,10 000,2), (14,14 000,3), (15,14 000,3)
6	(16,12 000,2)
7	(17,10 000,3), (18,15 000,2), (19,13 000,3)
8	(20,16 000,3), (21,16 000,3)
9	(22,14 000,3), (23,10 000,3), (24,16 000,2), (25,15 000,2)
10	(26,16 000,3), (27,14 000,2)

3.2　算法求解与结果分析

算法使用 Python 3.8.13 编程实现,在 AMD Ryzen 5 4600HV with Radeon Graphics,内存 8 GB,操作系统 Windows 10 环境下运行。

在本次实验中,设置 VNTS 参数为:最大迭代次数 $iter_{max}$ = 1 500, 当前最优解保持不变的最大次数 b_{max} = 250, 设置禁忌列表长度 L = 5, 候选集合的长度 CL = 10, 候选集中集中性和多样性元素的分界点长度 DL = 5。

利用 INNH 对初始可行解进行求解,根据不同的参数组合进行多组实验,输出目标值最优的解作为初始解,其对应的车辆配送方案和车辆配送信息如表 4 和 5 所示。本节表格中关于时间的单位为 h,关于成本的单位为元。

在初始解的基础上,分别使用传统 TS 和本文设计的 VNTS 进行优化求解从而获得近似最优解。两种算法求解结果对应的车辆配送方案和车辆配送信息如表 6 ~ 9 所示,算法的收敛曲线如图 4 所示,其中 TS 算法在第 1 176 次迭代获得最优解,VNTS 算法在第 269 次迭代获得最优解。

由表 6 ~ 9 可知,初始解对应的综合配送成本为 2 851.08 元,其中物流运输成本为 1 951.08 元,固定使用成本为 800.00 元,订单损失成本为 100.00 元;TS 获得的近似最优解对应的综合配送成本为 2 256.32 元,其中物流运输成本为 1 656.32 元,固

表 4　INNH 输出的车辆配送方案

车辆编号	行程	配送方案	路径
2	1	(1,2)	0-1-0
3	1	(2,14), (3,15), (1,26)	0-5-10-0
3	2	(1,27), (3,6), (2,11)	0-10-3-4-0
4	1	(1,25), (2,24), (3,22)	0-9-0
4	2	(3,20),(1,5),(2,4)	0-8-2-0
4	3	(3,10),(1,18),(2,19)	0-4-7-0
4	4	(3,21),(1,3),(2,1)	0-8-2-1-0
6	1	(2,13),(4,12),(1,16),(3,7)	0-5-6-3-0
6	2	(3,23),(1,8),(2,17),(4,9)	0-9-3-7-4-0

表 5　INNH 输出的车辆配送信息

车辆编号	配送时间/h	固定成本/元	运输成本/元	损失成本/元
2	2.14	200.00	270.66	0.00
3	6.86	200.00	598.77	0.00
4	10.18	200.00	621.97	100.00
6	6.06	200.00	459.68	0.00

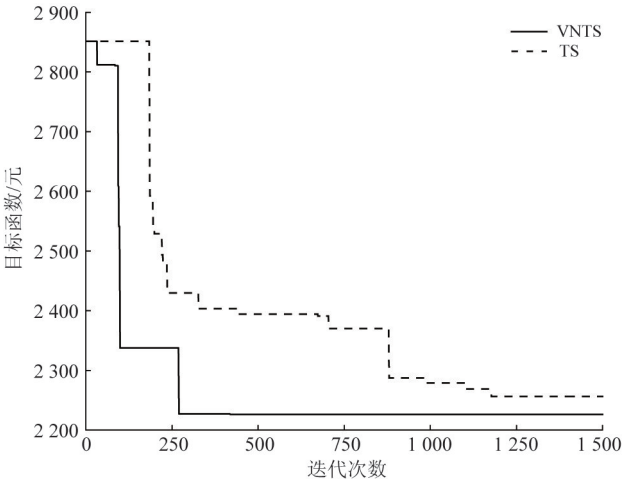

图 4　算法收敛曲线图

表 6　TS 输出的车辆配送方案

车辆编号	行程	配送方案	路径方案
3	1	(1,3)	0-2-0
3	2	(2,6),(3,8),(1,11)	0-3-4-0
4	1	(3,21),(1,5),(2,4)	0-8-2-0
4	2	(1,25),(2,24),(3,20)	0-9-8-0
4	3	(2,14),(1,27),(3,26)	0-5-10-0
4	4	(2,10),(1,18),(3,19)	0-4-7-0
6	1	(2,13),(4,15),(1,16),(3,7)	0-5-6-3-0
6	2	(1,12),(2,2),(4,1)	0-5-1-0
6	3	(2,23),(4,22),(3,17),(1,9)	0-9-7-4-0

表 7 VNTS 输出的车辆配送方案

车辆编号	行程	配送方案	路径方案
3	1	(1,6),(3,8),(2,11)	0-3-4-0
3	2	(2,14),(3,15),(1,26)	0-5-10-0
4	1	(3,20),(2,3)	0-8-2-0
4	2	(3,10),(1,18),(2,19)	0-4-7-0
4	3	(1,25),(2,24),(3,22)	0-9-0
4	4	(3,21),(1,5),(2,4)	0-8-2-0
6	1	(2,23),(3,17),(1,9)	0-9-7-4-0
6	2	(3,7),(4,27),(1,12)	0-3-10-5-0
6	3	(2,13),(1,16),(3,2),(4,1)	0-5-6-1-0

表 8 TS 输出的车辆配送信息

车辆编号	配送时间/h	固定成本/元	运输成本/元	损失成本/元
3	3.35	200.00	222.47	0.00
4	10.52	200.00	667.31	0.00
6	9.44	200.00	766.54	0.00

表 9 VNTS 输出的车辆配送信息

车辆编号	配送时间/h	固定成本/元	运输成本/元	损失成本/元
3	5.62	200.00	442.49	0.00
4	8.49	200.00	432.79	0.00
6	9.01	200.00	771.03	0.00

定使用成本为 600.00 元,订单损失成本为 0.00 元;VNTS 获得的近似最优解对应的综合配送成本为 2 226.31 元,其中物流运输成本为 1 626.31 元,固定使用成本为 600.00 元,订单损失成本为 0.00 元。

由于车辆数量有限,INNH 在求解时优先考虑完成加油站订单的配送需求,再考虑车辆配送路径的优化,因此初始解对应的车辆配送成本较大,也出现了订单部分未配送的情况。通过 TS 和 VNTS 的优化后,解的质量都有了较大的改进,TS 优化后的方案配送成本减少了 594.76 元(约占 20.9%),订单损失成本降低为 0.00 元,配送车辆由 4 辆车减少为 3 辆;VNTS 优化后的方案配送成本减少了 624.77 元(约占 21.9%),配送车辆由 4 辆车减少为 3 辆,订单损失成本降低为 0.00 元。综上所述,TS 和 VNTS 求解获得的车辆配送方案都有较好的可行性,相较之下,VNTS 获得的解的综合配送成本更低,且车辆工作时间更均衡,因此更加符合企业的需求。此外,观察图 4 可以发现,VNTS 收敛速度比传统的 TS 更快,因此具有更好的求解性能。

3.3 算法有效性分析

考虑到目前尚未有关于成品油配送的标准算例,本节在经典车辆路径问题的 Solomon 标准测试集的基础上,分别对测试集中 C 型、R 型、RC 型 3 类数据进行均匀采样,从而模拟加油站空间分布的聚集、离散、部分离散与部分聚集的场景,并以测试数据的配送中心作为油库、客户作为加油站,结合调研获得的油品需求特征随机生成加油站关于 92#、95#、98# 3 种油品的订单个数及容量,相应配送车辆及配送参数与 3.1 小节设置相同,最终构造出 24 组算例。其中,算例命名方式为 C1-S20-O60-V6,表示属于 C 型的第 1 个算例,其中包含 20 个站点和 60 个订单,配送车辆数为 6 辆。

为了验证本文设计的 VNTS 的求解性能,将其与商业求解器 Gurobi 以及 Stefan 等[23]提出的求解车辆路径问题的自适应大邻域搜索算法(adaptive large neighborhood search,ALNS)进行对比分析。为了使 ALNS 适用于本文问题的求解,基于 2.4 小节中所设计的邻域结构设置 ALNS 的邻域算子。此外,为了保证算法比较的公平性,VNTS 和 ALNS 均以同一个初始解为起点进行迭代,且终止条件设置相同。

对于每一组算例,先调用 Gurobi 求解器对问题的混合整数规划模型进行求解,记录求解器在最大允许时间内获得的最优解的目标值及运行时间;然后分别根据 ALNS 和 VNTS 对算例求解 10 次,记录并计算 10 次运行结果的最优值、平均值、标准差和平均运行时间。

本文通过 Python 接口调用 Gurobi 9.5.0,构建求解模型,设置 Gurobi 求解器最大求解时间为 1 800.00 s,精确解接受的最优解与当前解的差距 Gap 值为 0.001,VNTS 的参数设置与 3.2 小节相同。对 3 种方法关于不同算例的求解结果进行汇总,如表 10 所示。

在表10中,Obj表示算法和求解器求解获得的

表 10　算法求解表现比较

算例名称	Gurobi		ALNS				VNTS			
	Obj/元	*CPU*/s	*Obj*/元	*Mean*/元	*Std*	*CPU*/s	*Obj*/元	*Mean*/元	*Std*	*CPU*/s
C1-S05-O9-V6	1 501.67	0.97	1 501.68	1 521.31	51.72	1.69	1 501.68	1 508.10	24.42	3.15
C2-S05-O6-V6	1 454.22	0.51	1 454.22	1 473.94	18.30	0.69	1 454.22	1 459.00	12.39	1.58
C3-S10-O19-V6	2 238.15	27.86	2 438.15	2 449.95	19.33	2.13	2 291.30	2 432.71	35.17	4.15
C4-S10-O17-V6	2 812.60	23.00	2 925.28	3 054.14	62.52	1.72	2 856.10	2 995.15	54.42	3.65
C5-S15-O25-V6	3 799.55	1 800.00	4 107.08	4 299.29	101.86	2.09	3 876.57	4 003.55	70.98	3.97
C6-S15-O28-V6	3 947.28	1 800.00	4 150.02	4 433.20	155.57	2.74	3 880.00	4 014.49	24.88	4.88
C7-S20-O39-V6	5 914.60	1 800.00	6 373.25	6 925.91	179.85	3.42	5 856.45	6 183.60	124.09	6.04
C8-S20-O40-V6	7 419.02	1 800.00	7 878.35	8 385.54	224.37	3.04	7 148.99	7 471.47	169.58	5.79
R1-S05-O12-V6	1 394.85	4.68	1 572.32	1 638.16	18.77	1.06	1 439.50	1 614.64	59.52	2.09
R2-S05-O10-V6	1 707.05	0.95	1 788.12	1 816.85	54.75	1.37	1 788.12	1 791.95	8.70	2.54
R3-S10-O19-V6	2 004.55	27.35	2 004.55	2 085.19	42.30	2.40	2 004.55	2 053.71	31.20	4.59
R4-S10-O16-V6	2 097.40	7.47	2 197.40	2 209.10	21.55	1.94	2 197.40	2207.40	17.30	3.46
R5-S15-O28-V6	2 743.10	1 800.00	3 038.12	3 138.08	55.75	3.21	2 743.10	2 788.51	39.74	5.78
R6-S15-O29-V6	4 826.83	1 800.00	5 175.35	5 286.05	103.48	2.08	4 689.95	4 859.14	60.08	4.35
R7-S20-O38-V6	4 882.60	1 800.00	5 118.65	5 429.05	156.64	2.78	4 908.49	5 340.79	171.24	4.66
R8-S20-O40-V6	5 326.45	1 800.00	5 445.70	5 512.29	56.95	3.07	5 322.51	5 493.45	43.65	5.79
RC1-S05-O9-V6	1 116.40	0.80	1 116.40	1 245.82	36.23	1.21	1 116.40	1 241.58	23.64	2.43
RC2-S05-O10-V6	1 147.20	1.03	1 147.20	1 205.75	45.29	1.72	1 147.20	1 181.18	43.11	3.60
RC3-S10-O19-V6	2 851.38	7.33	3 361.82	3 361.82	0.00	1.05	2 895.48	3 310.96	106.06	2.26
RC4-S10-O21-V6	2 660.35	282.20	2 839.55	3 110.31	117.28	1.82	2 741.25	3 070.34	131.66	3.36
RC5-S15-O27-V6	2 790.70	1 800.00	2 963.65	3 187.95	102.61	2.59	2 803.40	2 909.11	46.85	5.84
RC6-S15-O29-V6	3 568.43	1 800.00	3 750.88	3 886.25	64.03	3.11	3 581.83	3 727.41	65.14	6.22
RC7-S20-O35-V6	4 909.27	1 800.00	4 841.70	5 104.40	177.34	3.04	4 780.94	4 857.87	48.77	5.72
RC8-S20-O38-V6	6 673.55	1 800.00	6 798.92	7 345.38	335.39	2.85	6 365.49	6 681.02	111.43	4.57

最优解的目标值(元),CPU 表示程序运行时间(s);Mean 和 Std 分别表示算法运行 10 次所得结果的平均值(元)和标准差。

由表 10 可以看出,在求解时间方面,对于较小规模的算例(5、10 个站点),Gurobi 求解器可以在较短时间内完成对全局最优解的求解,但随着需求站点和油品订单数量的增加,求解器计算时间呈指数级增长,且无法在可接受时间内获得精确解;而 ALNS 和 VNTS 面对较大规模的算例仍能实现对问题的快速求解,因此具有较好的实用性。

在最优解取值方面,参考 Gurobi 求解器计算结果,可知对于站点数为 5 和 10 的算例,VNTS 虽然不能都求解到问题的全局最优解,但算法获得的近似最优解与全局最优解已十分接近,其最大差距不超过 5%;对于站点数为 15 和 20 的算例,Gurobi 求解器在设置的最大求解时间(1 800.00 s)内已无法获取问题的精确解,而 VNTS 仍能快速求解问题模型,且有较大比例的求解结果比求解器给出的更优。因此随着问题规模的增大,VNTS 在求解性能上更具优势。相较于 VNTS,ALNS 虽然也可以在较短的时间内实现对问题的求解,但 ALNS 容易陷入局部最优,从而获得的最优解的质量明显低于 VNTS。

在求解稳定性方面,结合表中数据,可知 VNTS 运行 10 次所得的结果的平均值和标准差在总体上低于 ALNS,因此表明 VNTS 具有更好的稳定性,并且 VNTS 的求解质量优于 ALNS。

此外,针对 C 型、R 型、RC 型的算例,VNTS 都有较好的表现,说明算法在加油站离散分布、加油站聚集分布、部分加油站离散和部分加油站聚集分布的配送场景下都能有较好的应用。以上分析说明,对于 MCMTVRPLC 问题的求解,本文设计的 VNTS 具有较好的实用性与有效性。

4 结 论

本文结合成品油配送实际,提出了带装载组合约束的多舱位多行程的成品油二次配送车辆路径问题,以配送综合成本最低为目标构建了问题的混合整数规划模型,根据先分组后路径的求解策略设计了一种基于“移除-重插入”邻域结构的变邻域禁忌搜索算法,并设计了不同规模的算例实验验证了模型和算法的有效性。根据实验结果,可得出如下结论。

(1) 相较于 Gurobi 求解器,VNTS 算法能够在小规模算例下保证一定的求解精度,在较大规模算例下能够以远小于 Gurobi 求解器的求解时间获得质量更好的解。(2) 相较于文献[23]中的 ALNS,VNTS 获得的最优解的质量及求解性能更优。(3) VNTS 对于不同规模和不同站点分布情况下的成品油配送问题都有较好的适用性,能够提供较高质量的初始车辆配送方案,从而辅助调度人员进行相应计划与决策。考虑到成品油实际运输中往往存在油罐车运输以及加油站接受补货服务的时间窗,下一步将会对带时间窗约束的成品油配送车辆路径问题开展研究。

参考文献

[1] 蔡德洪,罗艳托. 成品油运输降费之道[J]. 中国石油石化,2014,(20):38-40.

[2] CORNILLIER F, BOCTOR F, LAPORTE G, et al. A heuristic for the multi-period petrol station replenishment problem [J]. European Journal of Operational Research, 2008,191(2):295-305.

[3] CORNILLIER F, LAPORTE G, BOCTOR F, et al. The petrol station replenishment problem with time windows[J]. Computers and Operations Research, 2009,36(3):919-935.

[4] BOCTOR F, RENAUD J, CORNILLIER F. Trip packing in petrol stations replenishment[J]. Omega, 2011,39(1):86-98.

[5] CORNILLIER F, BOCTOR F, RENAUD J. Heuristics for the multi-depot petrol station replenishment problem with time windows[J]. European Journal of Operational Research, 2012,220(2):361-369.

[6] 王旭坪,詹红鑫,孙自来,等. 多行程带补货时间窗的成品油多舱配送路径优化[J]. 管理工程学报, 2020,34(4):182-195.

[7] 高海龙,谢勇,马吉祥,等. 多行程多交货期的成品油配送优化[J]. 控制与决策, 2022,37(10):2714-2722.

[8] WANG L, KINABLE J, VAN WOENSEL T. The fuel replenishment problem: a split-delivery multi-compartment vehicle routing problem with multiple trips[J]. Computers and Operations Research, 2020,118:104904.

[9] 王莉. 成品油管网中转油库及成品油二次配送调度优化[D]. 北京:中国石油大学,2019.

[10] WEI X T, LIAO Q, ZHANG H R, et al. MILP formulations for highway petrol station replenishment in initiative distribution mode[J]. Petroleum Science, 2021,18(3):994-1010.

[11] 刘志硕,成屹男,左兴权,等. 考虑罐容限制的多油库多行程加油站配送优化问题研究[J]. 系统工程理论与实践, 2023, 43(4):1232-1248.

[12] 李珍萍,周文峰,张煜炜,等. 考虑卸载顺序约束的成品油二次配送车辆路径问题研究[J]. 控制与决策,2020,35(12):2999-3005.

[13] 李珍萍,杨光,韩倩倩. 考虑工作量均衡的成品油二次配送车辆路径问题[J]. 系统仿真学报,2022,34(2):221-233.

[14] 鲁建厦,李嘉丰,汤洪涛. 政府补贴和再制造收益影响下的集配一体化车辆路径问题研究[J]. 高技术通讯,2021,31(2):188-198.

[15] 李国明,李军华. 基于混合禁忌搜索算法的随机车辆路径问题[J]. 控制与决策,2021,36(9):2161-2169.

[16] 禹鑫燚,卢靓,朱熠琛,等. 基于启发式动态规划的自动化停车场调度方法[J]. 高技术通讯,2019,29(4):352-361.

[17] 姚冠新,范雪茹,张冬梅. 基于改进变邻域搜索的多隔室车辆路径优化算法[J]. 计算机集成制造系统,2022,28(9):2981-2997.

[18] 刘博,袁欣,明新国. 基于遗传算法和变邻域搜索的柔性作业车间同步调度方法[J]. 机械设计与研究,2021,37(1):177-182,189.

[19] 黄学文,陈绍芬,周阗玉,等. 求解柔性作业车间调度问题的一种新邻域结构[J]. 系统工程理论与实践,2021,41(9):2367-2378.

[20] CHRISTIAN P, PHILIPPE L, CAROLINE P. Order-first

split-second methods for vehicle routing problems: a review[J]. Transportation Research Part C: Emerging Technologies, 2014,40:179-200.

[21] SOLOMON M M. Algorithms for the vehicle routing and scheduling problems with time window constraints[J]. Operations Research, 1987,35(2):254-265.

[22] 贺一. 禁忌搜索及其并行化研究[D]. 重庆:西南大学,2006.

[23] STEFAN R, DAVID P. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows[J]. Transportation Science, 2006,40(4):455-472.

Optimization of routes for multi-compartment, multi-trip refined oil secondary distribution with loading combination constraints

WANG Cheng[*,**], YU Haonan[*], GUO Hengjun[*], ZHANG Wenzhu[*]

([*]College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023)

([**]Taizhou Research Institute, Zhejiang University of Technology, Taizhou 318001)

Abstract

The optimization problem of multi-compartment, multi-trip refined oil secondary distribution with loading combination constraints is addressed in this study. Heterogeneous multi-compartment oil tankers are utilized to transport different types of refined oil orders from oil depots to gas stations, with each order assigned to a single compartment. Loading multiple orders on the same vehicle must satisfy certain loading combination constraints. A mixed-integer programming model is established with the objective of minimizing the total distribution cost including the distribution cost of vehicles and the penalty cost for unfilled orders. To solve this model, a variable neighborhood tabu search heuristic algorithm based on the cluster-first, route-second strategy is proposed. Based on improving the initial solution constructed by the improved nearest-neighbor insertion algorithm, this algorithm optimizes the solution by applying 'removal-reinsertion' neighborhood operations to obtain an approximate optimal solution. Computational experiment results demonstrate that the proposed algorithm can effectively solve the problem within a short time. The proposed model and algorithm can provide theoretical support and decision-making guidance for refined oil sales enterprises in developing their secondary distribution schedules.

Key words: refined oil secondary distribution, multi-compartment, multi-trip, loading combination constraint, route optimization