

# 考虑订单取件时间和柔性时间窗的取送货车车辆路径问题

孙欣蕊, 李昆鹏, 刘腾博

(华中科技大学 管理学院, 湖北 武汉 430074)

**摘 要:**研究了同城配送中考虑订单取货时间和柔性时间窗的取送货车车辆路径问题, 考虑同城配送中订单起终点, 订单取货时间和订单配送的柔性时间窗, 车容量限制等因素。首先构建以配送成本与超时惩罚成本之和最小化为目标的混合整数线性模型。其次, 设计了含多种有效不等式及其对应分离算法的改进分支切割算法对该模型进行精确求解。最后通过实验测试分析了不等式的性能, 验证了算法的有效性, 实验表明适当的减少车辆数和增大装载能力能够有效的减少成本。

**关键词:**取送货车车辆路径问题; 有效不等式; 改进分支切割算法

**中图分类号:** O221; F252 **文章标识码:** A **文章编号:** 1007-3221(2022)07-0009-08 **doi:** 10.12005/orms.2022.0209

## Pickup and Delivery Vehicle Routing Problem with Order Release Dates and Flexible Time Windows

SUN Xin-rui, LI Kun-peng, LIU Teng-bo

(School of Management, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract:** This paper studies the pickup and delivery with order release dates and flexible time windows in the urban distribution, which considers the factors such as orders' pickup positions, orders' delivery positions, order release dates, flexible time windows and vehicle capacity. A mixed integer linear model is provided to minimize the sum of distribution cost and timeout penalty firstly. Secondly, an improved branch-and-cut algorithm considering a set of valid inequalities and corresponding separation algorithms is designed to solve the model accurately. Finally, we analyze the effect of inequality on the performance of the algorithm, and verify the effectiveness of the algorithm by testing several sets of examples. Moreover, the results also show that appropriately reducing the number of vehicles and increasing the loading capacity can effectively reduce the cost.

**Key words:** pickup and delivery vehicle routing problem; valid inequalities; improved branch-and-cut

## 0 引言

近年来, 同城配送市场不仅涌现一批新兴跑腿公司, 而且传统快递企业、外卖企业纷纷扩展同城跑腿业务, 从而同城跑腿配送已成为企业竞相争夺的蓝海市场。顺丰 2019 年同城业务营收 19.52 亿元, 同比增长 96.12%; 2020 年母亲节期间, UU 跑腿当日订单量增幅 50% 以上。由于我国同城配送市场仍处于行业发展初期, 各平台跑腿业务虽增速明显但配送模式尚未成熟。目前平台通常采用用

户在平台下单, 配送系统根据订单信息, 将订单分配至骑手, 待取件后平台自动生成预计到达时间再由骑手完成配送的模式。但因用户在下单时无法设置预期送达时间, 导致订单完成时间多取决于骑手, 存在很大不确定性, 故此配送模式不适用于客户希望在指定时间内送达, 骑手尽可能多地接收订单的实际场景。因此, 基于此场景, 如何科学的制定配送方案, 进而提高配送效率、节约运力资源是需深入研究的重要课题。

本文所考虑的取送货配送模式: 首先, 用户在平台下单, 每个订单信息包括任务起点和终点的具

收稿日期: 2020-07-05

**基金项目:** 国家自然科学基金重点国际合作项目 (71620107002); 湖北省普通高等学校人文社会科学重点研究基地 - 企业决策支持研究中心 (DSS20200706)

**作者简介:** 孙欣蕊 (1991-), 女, 河南濮阳人, 博士研究生, 研究方向: 车辆路径优化; 李昆鹏 (1978-), 男, 通讯作者, 湖北武汉人, 教授, 研究方向: 供应链与物流管理、生产运作管理; 刘腾博 (1997-), 河南郑州人, 硕士研究生, 研究方向: 智能物流调度。

体位置、物品属性、取件时间限制及送达时间窗要求;然后,配送中心对一定时间内的客户订单,根据订单的取件时间要求及起终点位置为骑手派单并规划路线。在此配送模式下,为保障骑手权益,系统根据起终点位置之间的距离设定合理送达时间窗限制;为提升客户满意度,若骑手在送达时间窗外延迟或提前将物品送达,平台将根据偏离时间窗的程度采取一定惩罚,即用户预期送达时间窗为柔性时间窗约束且具有一定的惩罚成本,平台总惩罚成本越低说明配送准时率及客户满意度越高。基于此,本文以配送成本和惩罚成本之和最小为目标,研究具有订单取件时间和柔性时间窗约束的取送货车车辆路径问题(PDVRPORDFTW)。

具有不同约束条件的取送货问题一直是研究热点,常见的约束条件包括:客户需求特点、车容量、车辆限制等。同时取送货车车辆路径问题(VRP-SPD)(马艳芳等<sup>[1]</sup>);需求可拆分的取送货问题(VRPSPD)(李寒梅<sup>[2]</sup>);带时间窗的取送货车问题(VRPTW)(祁文祥等<sup>[3]</sup>,边展等<sup>[4]</sup>)。本文考虑实际客户需求,研究具有订单取货时间和柔性时间窗约束的取送货车车辆路径问题(PDVRPRDFTW),目前尚未有学者对具有两个客户时间约束的车辆路径问题进行探讨,相关研究多集中在带硬时间窗的取送货问题(PDPTW)以及带软时间窗的取送货问题(PDPSTW)。很多学者针对PDPTW提出精确算法和启发式算法进行求解。其中精确算法包括:分支-定价算法(Sun等<sup>[5]</sup>;Ghilas等<sup>[6]</sup>)。分支切割算法(Ropke等<sup>[7]</sup>,Furtado等<sup>[8]</sup>)。分支-切割-定价算法(Ropke等<sup>[9]</sup>),动态规划(Mahmoudi等<sup>[10]</sup>),基于集合分割的精确算法(Baldacci等<sup>[11]</sup>)。相关启发式算法包括:禁忌搜索算法(Nanry等<sup>[12]</sup>;Goeke<sup>[13]</sup>),自适应大邻域搜索算法(Ropke等<sup>[14]</sup>),模拟退火算法(Li等<sup>[15]</sup>)。祁文祥等<sup>[3]</sup>对PDPSTW提出节约算法进行求解。分析发现,极少有学者对PDPSTW问题展开研究;对PDPTW问题的研究,集中在算法性能的优化及求解规模的提升。此外,已有文献在构造数学模型时多考虑车容、取货或送货的时间窗、访问优先级等约束,显然无法适用于客户对取货和送货时间均有要求的同城配送场景。基于此,本文在构建模型时将客户的取货和送货时间均考虑在内,并设计改进分支切割算法。相较于现有研究,更符合同城配送平台跑腿业务的实际需求,并且采用精确算法得到的最优解可以用来评价启发式算法的有效性,具有一定的指导作用。

本文的主要贡献包含:(1)构建了混合整数模型,并将非线性约束进行线性化转换;(2)根据问

题的特性,考虑四种有效不等式,设计改进分支切割算法对模型进行求解;(3)根据实际数据构造算例,验证算法的有效性。

## 1 数学模型

### 1.1 问题描述及假设

本文所研究的问题描述如下:针对多个订单,配送企业进行订单分配和规划配送路线。已知信息包括:每个订单相应的起点和终点地址信息、订单需求量、每个订单对应的起点有相应的订单取件时间;每个订单对应的终点有相应的服务时间窗。客户对车辆的提早或延迟到达有一定的容忍度,因此考虑柔性的服务时间窗。目标是配送成本和惩罚成本之和最小。为了方便建模,考虑以下假设:(1)配送车辆数固定、车辆同质且匀速行驶、电量充足;(2)每个订单对应的需求量不超过车辆的装载能力;(3)每个订单由同一车辆进行配送;(4)车辆允许提前到达订单终点,但需在相应的时间限制内服务客户。(5)车辆服务客户的时间超出系统预计的时间窗,但未超出最晚到达时间时,有相应的惩罚。

### 1.2 符号说明

$R$ :客户订单集合  $R = \{1, \dots, n\}$ ;  $P = \{1, \dots, n\}$ :订单起点集合,第  $i$  个订单对应第  $i$  个起点;  $D = \{n+1, \dots, 2n\}$ :订单终点集合,第  $i$  个订单对应第  $n+i$  个终点;  $N = P \cup D \cup \{0, 2n+1\}$ :节点集合,0 和  $2n+1$  均表示配送站;  $A$ :弧的集合,  $A = \{(i, j) \mid i \in N, j \in N\}$ ;  $V$ :车辆集  $V = \{1, \dots, K\}$ ;  $c$ :单位距离运输成本;  $C$ :单位时间延迟成本;  $T$ :车辆的最长工作时间;  $d_{ij}$ :从节点  $i$  到节点  $j$  的行驶距离;  $t_{ij}$ :车辆服务节点  $i$  的时间与从节点  $i$  到节点  $j$  的行驶时间之和;  $Q$ :车辆的最大运载量;  $ET_i$ :订单起点  $i$  的最早可取件时间;  $(q_i + q_{n+i})$ :  $q_i$  和  $q_{n+i}$  分别表示订单起点  $i$  和终点  $n+i$  的配送量,  $q_i = -q_{n+i} \geq 0$ ;  $[ET_{n+i}, LT_{n+i}]$ :订单终点  $n+i$  的预期送达时间窗;  $ELT_{n+i}$ :订单终点  $n+i$  的最迟送达时间;  $x_{ij}$ :若车辆从节点  $i$  行驶至节点  $j$ ,则为 1;否则为 0;  $A_i$ :车辆到达节点  $i$  的时间;  $B_i$ :车辆开始服务节点  $i$  的时间;  $Q_i$ :车辆访问节点  $i$  后的车载量;  $v_i$ :表示节点  $i$  的路径标志符号。

### 1.3 数学模型

#### 1.3.1 模型建立

$$\min \sum_{i \in N} \sum_{j \in N} cd_{ij}x_{ij} + \sum_{n+i \in D} P(B_{n+i}) \tag{1}$$

$$\sum_{j \in P} x_{0j} = |V| \tag{2}$$

$$\sum_{i \in N} x_{ij} = 1, \forall j \in P \cup D \quad (3)$$

$$\sum_{i \in N} x_{ji} = 1, \forall j \in P \cup D \quad (4)$$

$$\sum_{i \in D} x_{i,2n+1} = |V| \quad (5)$$

$$Q_i \leq \min\{Q, Q + q_i\}, \forall i \in P \cup P \quad (6)$$

$$Q_i \geq \max\{0, q_i\}, \forall i \in P \cup P \quad (7)$$

$$Q_i + q_j - Q_j - Q(1 - x_{ij}) \leq 0 \\ \forall i \in N \setminus \{2n+1\}, j \in N \setminus \{0\} \quad (8)$$

$$B_i - A_i \geq 0, \forall i \in P \cup D \quad (9)$$

$$B_j \geq B_i + t_{ij} - T(1 - x_{ij}), \forall i, j \in P \cup D \quad (10)$$

$$B_{n+1} - B_i \geq t_{i,n+1}, \forall i \in P \quad (11)$$

$$A_j \geq B_i + t_{ij} - T(1 - x_{ij}) \\ \forall i \in P \cup D \cup \{0\}, \forall j \in P \cup D \quad (12)$$

$$A_j \leq B_i + t_{ij} + T(1 - x_{ij}) \\ \forall i \in N \setminus \{2n+1\}, \forall j \in P \cup D \quad (13)$$

$$B_j = \max\{A_j, ET_j\}, \forall j \in P \quad (14)$$

$$B_{n+j} = \max\{A_{n+j}, ET_{n+j}\}, \forall n+j \in D \quad (15)$$

$$B_{n+j} \leq ELT_{n+j}, \forall n+j \in D \quad (16)$$

$$P(B_j) = \begin{cases} 0, & ET_j \leq B_j \leq LT_j \\ C(B_j - LT_j), & LT_j < B_j \leq ELT_j \end{cases},$$

$$\forall j \in D \quad (17)$$

$$1 \leq v_i \leq n, \forall i \in P \cup D \quad (18)$$

$$v_i \geq ix_{0i}, \forall i \in P \quad (19)$$

$$v_i \leq n - (n - i)x_{0i}, \forall i \in P \quad (20)$$

$$v_{n+i} - v_i = 0, \forall i \in P \quad (21)$$

$$v_i - v_j \leq n(1 - x_{ij}), \forall i, j \in P \cup D \quad (22)$$

$$v_i - v_j \geq n(1 - x_{ij}), \forall i, j \in P \cup D \quad (23)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in N \quad (24)$$

$$v_j \in \{1, \dots, n\}, \forall j \in P \cup D \quad (25)$$

$$A_i \geq 0, \forall i \in P \cup D \quad (26)$$

$$B_i \geq 0, \forall i \in P \cup D \quad (27)$$

$$0 \leq Q_j \leq Q, \forall j \in N \quad (28)$$

其中(1)为目标函数,使配送成本和惩罚成本之和最小。(2)表示有  $IN$  辆车从配送中心出发,执行配送任务,车辆访问的第一个节点是订单对应的起点。(3)~(4)表示每个节点被访问一次,并保证流平衡。(5)保证所有车辆配送访问的最后一个节点是配送站。(6)~(7)表示车辆访问节点后,车辆装载量的限制。(8)保证车辆装载量的一致性。(9)表示车辆到达节点后才开始服务。(10)保证车辆连续先后访问节点  $i$  和节点  $j$  时,  $B_j \geq B_i + t_{ij}$ 。(11)表示一个订单只能由一辆车配送,并且车辆需先到订单对应的起点取货,再配送至订单对应的终点。(12)~(13)保证当车辆先后连续访问两个节点  $i$  和  $j$  后,  $A_j = B_i + t_{ij}$ 。(14)表示当车辆到达订单对应起点的时间小于等于它对应订单取

件时间,则需在订单取件时间取货。(15)表示当车辆提前到达订单对应终点,需要等待,直到订单对应终点预期的最早服务时间点,才进行服务。(16)表示车辆需在订单对应终点可承受的最晚送达时间点前服务。(17)表示当配送服务顾客的时间点超过预期时间窗,但不超过最迟服务时间点时,有一定的惩罚。(18)~(23)表明同一车辆访问的所有节点的路径标志符号相同,且路径标志符号等于该车辆访问的第一个节点序号。(18)规定了所有节点的路径标志号的取值范围。(19)~(20)保证当车辆离开配送站后访问的第一个节点为时,节点  $i$  的路径标识符号为  $i$ 。(21)保证单个订单的取货点和送货点由同一车辆进行访问。(22)~(23)表示如果车辆连续访问节点  $i$  和  $j$  时,则节点  $i$  和  $j$  的路径标识符是相同的。(24)~(28)是定义的变量取值限制。

### 1.3.2 模型的线性化

由于约束(14)、(15)、(17)是非线性化形式,因此上述模型是非线性的。为降低模型求解的复杂度,将(14)(15)(17)转化为线性形式。

为使(14)~(15)线性化,定义 0-1 变量,  $u_{0i}$ ,  $\forall i \in P$  并用以下约束(29)~(34)代替:

$$B_i \geq A_i, \forall i \in P \cup D \quad (29)$$

$$B_i \leq A_i + T(1 - u_{0i}), \forall i \in P \cup D \quad (30)$$

$$B_i \geq ET_i, \forall i \in P \cup D \quad (31)$$

$$B_i \leq ET_i + Tu_{0i}, \forall i \in P \cup D \quad (32)$$

$$A_i \geq ET_i - T(1 - u_{0i}), \forall i \in P \cup D \quad (33)$$

$$A_i \leq ET_i + Tu_{0i}, \forall i \in P \cup D \quad (34)$$

为线性化(17),定义变量  $w_{n+i,0}, w_{n+i,1}, w_{n+i,2}$ ,  $z_{n+i,0}, z_{n+i,1}$  并用(35)~(43)代替:

$$w_{n+i,0}, w_{n+i,1}, w_{n+i,2} \geq 0, \forall n+i \in D \quad (35)$$

$$z_{n+i,0}, z_{n+i,1} \in \{0, 1\}, \forall n+i \in D \quad (36)$$

$$w_{n+i,0} + w_{n+i,1} + w_{n+i,2} = 1, \forall n+i \in D \quad (37)$$

$$z_{n+i,0} + z_{n+i,1} = 1, \forall n+i \in D \quad (38)$$

$$w_{n+i,0} \leq z_{n+i,0}, \forall n+i \in D \quad (39)$$

$$w_{n+i,1} \leq z_{n+i,0} + z_{n+i,1}, \forall n+i \in D \quad (40)$$

$$w_{n+i,2} \leq z_{n+i,1}, \forall n+i \in D \quad (41)$$

$$B_{n+i} = ET_{n+i} w_{n+i,0} + LT_{n+i} w_{n+i,1} + ELT_{n+i} w_{n+i,2}, \forall n+i \in D \quad (42)$$

$$P(B_{n+i}) = C(ELT_{n+i} - LT_{n+i}) w_{n+i,2} \quad (43)$$

从而为使目标(1)线性化,考虑

$$\max \sum_{i \in N} \sum_{j \in N} cd_{ij} x_{ij} + \sum_{n+i \in D} C(ELT_{n+i} - LT_{n+i}) w_{n+i,2} \quad (44)$$

因此,混合整数线性模型为(44),(2)~(13),(16),(18)~(43)。在下述部分中是考虑该线性模型。

## 2 有效不等式

为使模型在规定时间内获得最优解或紧的下界,本节根据问题特性,提出一些有效不等式来提高模型的性能。为此,先定义以下符号: $S \subset (P \cup D)$  为节点子集,  $\bar{S} = N \setminus S$  是  $S$  的余集;  $\rho(S) = \{i \in P \cap S \mid n+i \notin S\}$  和  $\gamma(S) = \{n+j \in S \cap D \mid j \notin S\}$  是  $S$  的子集;  $\pi(S) = \{i \in P \mid \forall n+i \in S\}$  是  $S$  的前任集合;  $\sigma(S) = \{n+i \in D \mid i \in S\}$  是  $S$  的后任集合。

### 2.1 车容量不等式

对于  $AS \subset P \cup D$ ,  $k(S)$  表示服务集合  $S$  所需车辆数的下界。在任意可行解中,服务  $S$  中所有节点的的车辆数大于等于  $k(S)$ , 其中,  $k(S)$  可取值为  $\max\{1, \lceil q(\pi(S) \setminus S)/Q \rceil, \lceil -q(\sigma(S) \setminus S)/Q \rceil\}$  [9], 则可表示为:

$$\sum_{i \in N \setminus S, j \in S} x_{ij} \geq k(S) \quad (45)$$

因车辆的装载量不超过  $Q$ , 可考虑集合  $S = (S \cap P) \cup (S \cap D)$ , 对  $\forall i \in S \cap P$ , 令  $U_i = \{j \in P \cap \bar{S} \mid q_i + q_j > Q\}$ , 则  $\{(i, j) \mid j \in U_i\}$  不包含在任意可行解中; 对  $\forall n+j \in S \cap D$ , 令  $W_{n+j} = \{i \in P \cap \bar{S} \mid q_i - q_{n+j} > Q, i \neq j\}$ ,  $R_{n+j} = \{n+h \in D \cap \bar{S} \mid q_{n+i} + q_{n+j} < -Q\}$ , 则  $\{(i, n+j) \mid i \in W_{n+j} \cup R_{n+j}\}$  不包含在任意可行解中。因此, 式(45)可提升为以下不等式:

$$\sum_{i \in S, j \in S} x_{ij} - \sum_{i \in S \cap P, j \in U_i} x_{ji} - \sum_{i \in S \cap D, j \in W_i} (\sum_{j \in W_i} x_{ji} + \sum_{j \in R_i} x_{ji}) \geq k(S) \quad (46)$$

### 2.2 优先顺序不等式

考虑集合  $S \subset N (0 \in S, 2n+1 \notin S)$ , 至少存在节点  $j \in P$  满足:  $j \in \bar{S}; n+j \in S$ 。则任意一个可行解满足以下不等式:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 2 \quad (47)$$

这是因为本文研究的问题存在以下事实: 车辆从配送站  $o$  出发, 完成配送任务后, 返回配送站  $2n+1$ ; 单个订单对应的起点和终点由同一车辆访问; 车辆满足优先约束, 即车辆访问  $n+j$  之前, 需先访问订单对应的起点  $j$ 。假如  $\sum_{i \in S} \sum_{j \in S} x_{ij} = |S| - 1$ , 由于每个节点当且仅当被访问一次, 则  $S$  中所有节点形成子循环路线。这与上述事实矛盾, 因此(47)成立。

同理, 考虑集合  $T \subset N (0 \notin T, 2n+1 \in T)$ , 至少存在一个取货点  $i \in T$  满足:  $i \in T, n+i \notin \bar{T}$ 。则下列不等式是有效的:

$$\sum_{i \in T} \sum_{j \in T} x_{ij} \leq |T| - 2 \quad (48)$$

### 2.3 加强子回路移除不等式

考虑集合  $S \subset P \cup D$ , 则任意一个可行解满足

以下不等式:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad (49)$$

这是因为若  $\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S|$ , 则  $S$  的节点构成子回路, 这将违反车辆从配送中心  $0$  处出发, 开始配送的事实。因此,  $\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1$ 。故式(49)成立。

基于  $S$ , 考虑  $\forall p \in \rho(S), \forall n+q \in \gamma(S)$ , 则式(49)可提升为下述有效不等式:

$$x_{n+p,q} + x_{q,n+p} + \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad (50)$$

这是因为假设  $x_{n+p,q} + x_{q,n+p} + \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S|$  由于任意可行解均满足  $x_{n+p,q} + x_{q,n+p} \leq 1, \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1$ , 则有  $x_{n+p,q} + x_{q,n+p} = 1, \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1$ 。

根据  $\sum_{i \in S} \sum_{j \in S} x_{ij} = |S| - 1$  可知, 存在节点  $p$  到节点  $n+q$  的路径或者从节点  $n+q$  到节点  $p$  的路径, 这导致  $x_{n+p,q} = 0, x_{q,n+p} = 1$ , 因而产生矛盾。故式(50)成立。

### 2.4 路径不可行不等式

**命题 1**  $W = (n+i, j, n+k)$  表示包含三个节点的路径, 其中  $n+i, n+k \in D, j \in P$ 。令  $W_1 = (i, k, n+i, j, n+k, n+j)$ ,  $W_2 = (k, i, n+i, j, n+k, n+j)$ 。若  $W_1$  和  $W_2$  均不可行, 则路径  $W$  不包含在任意可行解中。

**证明** 假设  $W$  存在一个可行解中, 因车辆配送满足优先约束, 则节点  $i$  和节点  $k$  在  $W$  之前被访问, 节点  $n+j$  在  $W$  之后被访问。然而, 与假设条件矛盾, 因此  $W$  不包含在任意可行解中。

**命题 2** 对于节点  $i \in P$ , 有  $q_i + q_j > Q, \forall j \in P \setminus \{i\}$ 。令  $R = \{0, j, n+j, i\}$ , 其中  $i, j \in P, n+j \in D$ 。如果对任意  $j \in P, R$  不可行, 则节点  $i$  是车辆离开节点  $0$  后访问的第一个节点。

**证明** 假设节点  $i$  不是车辆离开节点  $0$  后访问的第一个节点, 且对于任意节点  $j \in P \setminus \{i\}$ , 有  $q_i + q_j > Q$ , 从而节点  $h \in P \setminus \{i\}$  和  $n+h \in D$  满足  $0 \rightarrow h \rightarrow n+h \rightarrow i$ 。然而, 与假设条件矛盾, 因此节点  $i$  是车辆离开节点  $0$  后访问的第一个节点。

综上, 则下述不等式有效:

$$x_{n+i,j} + x_{j,n+k} \leq 1, W_1, W_2 \text{ 不可行} \quad (51)$$

$$x_{0,i} = 1, R \text{ 不可行} \quad (52)$$

## 3 改进分支切割算法

### 3.1 预处理

若任意可行路径不包含弧  $(i, j) \in A$ , 则称  $(i, j)$

是冗余的。因此,为避免出现冗余的情况,通过在数学模型中加入以下约束,加快求解速度。

(1)对于满足条件  $ET_{n+i} > ELT_{n+i}$  的送货点,有  $n+i, n+j \in D$ , 有  $x_{n+i, n+j} = 0$ 。

(2)如果  $P = (j, i, n+j, n+i)$  为不可行路线,则弧  $(i, n+j)$  是冗余的,从而  $x_{i, n+j} = 0$ ; 同理,如果  $P = (i, n+i, j, n+j)$  为不可行路线,则弧  $(n+i, j)$  是冗余的,从而  $x_{n+i, j} = 0$ 。

(3) 2.4 部分的路径不可行不等式 (51), (52)。

### 3.2 分离算法

令  $z^*$  为搜索树中节点对应线性规划松弛的非整数目标值,则存在相应的加权有向图  $G^* = (N, A^*)$ , 其中  $*$  =  $\{(i, j) | x_{ij}^* > 0\}$ ,  $G^*$  中每一条弧  $(i, j)$  的权值为  $x_{ij}^*$ 。基于  $G^*$  搜索违反第2部分三种不等式的方法如下。

#### 3.2.1 分离车容量不等式

令  $S \subset P \cup D, f(S) = \sum_{i \in S} \sum_{j \in S} x_{ij}^* - k(S)$ ,  $F$  是  $f(S)$  的集合。根据  $G^*$ , 先采用构造启发算法,寻找违反的容量不等式。构造启发式法:首先从集合  $P \cup D$  中随机选取一个节点放入  $S$  中,然后对下述操作重复迭代30次。对于集合  $(P \cup D) \setminus S$  中任意点  $i$ , 令  $S' = S \cup \{i\}$ , 计算对应  $f(S')$ , 并将  $f(S)$  添加至  $F$  中。对于  $S$  中任意节点  $j$ , 令  $S'' = S \setminus \{j\}$ , 计算对应的  $f(S'')$ , 并将加入到  $F$  中。最后更新  $S'''$  为  $F$  中最小值对应的节点集合,如  $f(S''') \leq -0.1$ , 则确定一个违反的容量不等式,令  $S = S'''$ 。

#### 3.2.2 分离优先不等式

根据有向图  $G^*$ , 对于每个节点  $n+i \in D$ , 遍历从点0到点  $n+i$  的所有路径。对于每条路径,令  $S$  为该路径对应的节点集合,并且  $f(S) = \sum_{i \in S} \sum_{j \in S} x_{ij}^* - (|S| - 2)$ 。如果对应的  $f(S) > 0.1$ , 则确定一个违反的优先不等式。同理,对于每个节点  $i \in P$ , 遍历从节点  $i$  到节点  $2n+1$  的所有路径。对于每条路径,令  $T$  为该路径对应的节点集合,并且  $f(T) = \sum_{i \in T} \sum_{j \in T} x_{ij}^* - (|T| - 2)$ 。如果对应的  $f(T) \geq 0.1$ , 则确定一个违反的优先不等式。

#### 3.2.3 分离子回路移除不等式

根据图  $G^*$ , 考虑禁忌搜索算法寻找违反的子回路移除不等式。对于节点集合  $S$ ,  $f(S) = \sum_{i \in T} \sum_{j \in T} x_{ij}^* + \sum_{i \in \pi(\gamma(S))} \sum_{j \in \sigma(\rho(S))} x_{ij}^* - (|S| - 1)$ , 从集合  $P \cup D$  中随机选取一个送货点添加到  $S$  中。令  $Tabu$  表示禁忌表,设置禁忌对象为取货点,禁忌长度为5,最大迭代次数为30,在每次迭代中可从以下两

种操作中选择任意一种。(1)从  $\bar{S} \setminus \{0, 2n+1\}$  中选取节点  $i \in P \cup D$ , 使  $f(S \cup \{i\})$  值最大。如果节点  $i$  不在  $Tabu$  中,则将其添加到  $S$  中,并判断是否  $f(S) > 0.1$ , 若是,则确定一个违反的子回路移除不等式。从  $S$  中删除节点  $i$ , 并将其放入  $Tabu$  中以此更新禁忌表。(2)从  $S$  中选取节点  $j$ , 使  $f(S \setminus \{j\})$  值最大。若节点  $j$  不在  $Tabu$  中,判断是否  $f(S \setminus \{j\}) \geq 1$ , 若是,则确定一个违反的子回路移除不等式。从  $S$  中删除节点  $j$ , 并将其放入  $Tabu$  中以此更新禁忌表。

### 3.3 改进分支切割算法求解步骤

**步骤1 初始化:**令  $\Phi$  表示活跃节点的集合,当前活跃节点  $t$  为根节点0,将其添加到  $\Phi$ ;  $z_{best}$  为目标最优解,  $z_{best} \leftarrow +\infty$ 。

**步骤2 预处理:**根据3.1部分的预处理,删除图  $G$  中冗余的弧,从而减少求解规模。

**步骤3 求解松弛的线性规划模型,**包含约束 (44), (2) ~ (13), (16), (18) ~ (23), (26) ~ (43), 和  $0 \leq x_{ij} \leq 1, \forall i, j \in N, 1 \leq v_y \leq n, \forall y \in P \cup D$  得到解  $z_t^*$ 。

**步骤4 若  $z_t^*$  为不可行解,**则令  $\Phi \leftarrow \Phi \setminus \{t\}$ , 转至步骤7。

**步骤5 若  $z_t^*$  为整数可行解,且  $z_t^* < z_{best}$ ,**则令  $z_{best} = z_t^*, \Phi \leftarrow \Phi \setminus \{t\}$  转至步骤7。

**步骤6 若  $z_t^*$  为小数可行解,**当前活跃节点  $t$  为根节点时,分别执行3.2部分的分离算法确定违反的不等式;当前活跃节点  $t$  为子节点时,执行容量不等式和优先不等式的分离算法寻找违反的不等式。如果存在违反的不等式,则将它们添加到模型中,重新求解,返回步骤4;如果不存在违反的不等式,则转至步骤8。

**步骤7 节点选择:**若  $\Phi$  是空集,则算法停止;若  $\Phi$  不是空集,则根据下界优先策略选择一个节点  $t$  进行求解,令  $\Phi \leftarrow \Phi \setminus \{t\}$ , 并转到步骤4。

**步骤8 分支:**从  $\{x_{ij} | \forall i, j \in N\}$  中选取一个对应的解是小数解的变量进行分支,生成两个子节点,并添加到  $\Phi$ 。

## 4 数值实验

### 4.1 算例构造

根据某平台的订单数据,提取预约模式下某时段内的订单数据。利用百度拾取坐标系统,计算任

意两个节点之间的距离。在此基础上构造 5 组不同订单规模的算例进行测试  $\{10, 20, 30, 40, 50\}$ ，每种规模随机生成 5 个算例，共 25 个算例，各项实验参数如下：车辆装载能力  $Q = 7$ ；车辆最长工作时间  $T = 60$ ；单位配送成本  $c = 0.2$ ；单位时间延迟成本  $C = 0.2$ ；任意两节点间的行驶时间  $t_{ij} = d_{ij}/speed$  ( $speed = 21\text{km/h}$ )；客户节点的时间窗长度  $W = 25$ ；对每个订单  $(i, n + i)$ ，对应的需求量  $q_i \sim U[1, 7]$ ， $q_{n+i} = -q_i, r_i \sim U[5, 10]$ ， $ET_{n+i} = r_i + t_{i, n+i}$ ， $LT_{n+i} = ET_{n+i} + W$ ； $ELT_{n+i} = LT_{n+i} + 5, q_i \sim U[1, 7]$ 。

4.2 有效不等式性能分析

为分析提出的有效不等式对模型的改进效果，考虑在搜索树的根节点处添加不同的有效不等式，并比较它们对下界的影响。表 1 表示不同的不等式添加策略下的下界值。第 1 列为算例名称（例

如 10\_1, 10 个节点规模下第 1 个算例）；第 2 列为只求解模型得到的下界值；Prece、Subtour、Cap 分别表示优先顺序不等式，加强子回路移除不等式和车容量不等式。第 3~8 列是在搜索树的根节点处添加对应的不等式的下界值。

结果表明，对于 8 种处理方式下的平均下界，在搜索树的根节点处，相比于只求解模型，7 种处理方式均能不同程度的提高搜索树根节点的下界；同时考虑 3 种有效不等式具有最好的平均下界。此外，相比于优先顺序不等式和加强子回路不等式，车容量不等式的效果最好；在搜索树的根节点处，相比于考虑一种不等式或两种不等式的添加策略，同时考虑三种不等式是最好的不等式添加策略。因此，在改进分支切割算法中，将同时考虑三种不等式作为添加策略。

表 1 有效不等式的结果比较

算例	Model	Prece	Subtour	Cap	Prece + Subtour	Prece + Cap	Subtour + Cap	Prece + Subtour + Cap
10_1	14. 6	15. 04	14. 89	14. 67	14. 89	15. 08	14. 95	15. 08
20_1	20. 48	21. 13	21. 16	22. 2	21. 22	22. 25	22. 34	22. 45
30_1	45. 07	45. 14	45. 11	46. 66	45. 19	46. 85	46. 68	46. 67
40_1	67. 11	67. 3	67. 11	67. 75	67. 25	67. 86	67. 74	67. 91
50_1	104. 18	104. 25	104. 26	104. 26	104. 28	104. 3	104. 27	104. 32
平均值	50. 29	50. 57	50. 5	51. 12	50. 57	51. 27	51. 2	51. 29

4.3 改进分支切割算法与 CPLEX 默认的分支切割算法的比较

为验证改进分支切割算法的效果，将其与 CPLEX 默认的分支切割算法进行比较。表 2 表示时间限制为 1h 下两种算法求解的结果。其中第 1 列与表 1 中的第一列有相同的含义。统计两种算法下每个算例的下界，运行时间，Gap。Gap = 100% (第 5 列值-第 3 列值)/第 3 列值，“\*”表示该算例求得最优解。

由表 2 可知，(1)对于两种算法获得最优解的个数，25 个算例中，改进分支切割算法能够获得 25 个算例的最优解；CPLEX 默认的分支切割算法只能够获得 6 个算例的最优解。(2)对于两种算

法在 1h 的运行时间内获得的平均下界，改进分支切割算法的平均下界为 50.94，CPLEX 默认的分支切割算法的平均下界为 38.08，故相比于 CPLEX 默认的分支切割算法，改进分支切割算法使平均下界提高 33.8%。(3)对于不同规模的算例，两种算法对规模为 10, 20, 30, 40, 50 的算例，改进分支切割算法分别求得 5, 5, 5, 5, 5 个算例的最优解；CPLEX 默认的分支切割算法分别求得 5, 1, 0, 0, 0 个最优解。综上，算例规模较小时，改进分支切割算法求解时间短，有实用性；随着规模的增加改进分支切割算法可以在较短时间内求得最优解；并可代替 CPLEX，提供更好的下界，评估相关问题的启发方法的性能。

表 2 改进分支切割算法与 CPLEX 默认分支切割算法的结果比较

算例	车辆数	CPLEX 默认分支切割算法		改进分支切割算法			下界
		下界	时间(秒)	下界	时间(秒)	cuts 数	Gap(%)
10_1	2	15.08 *	5.7	15.08 *	1	37	0
10_2	2	12.37 *	15.6	12.37 *	1	38	0
10_3	2	8.06 *	16	8.06 *	1	16	0
10_4	2	10.23 *	17	10.23 *	1	33	0
10_5	2	9.41 *	17	9.41 *	1	58	0

20_1	2	22.59 *	764	22.59 *	15	213	0
20_2	2	17.29	3600	19.38 *	1203	317	12.09
20_3	2	12.72	3600	14.39 *	31	389	13.13
20_4	2	19.47	3600	20.07 *	145	243	3.08
20_5	2	18.64	3600	21.65 *	618	323	16.15
30_1	3	40.6	3600	47.76 *	184	483	17.64
30_2	3	30.34	3600	42.02 *	924	399	38.5
30_3	3	33.51	3600	40.87 *	1005	302	21.96
30_4	3	38.21	3600	45.43 *	6	234	18.9
30_5	3	38.48	3600	48.52 *	479	289	26.09
40_1	5	55.13	3600	68.91 *	1105	410	25
40_2	5	45.19	3600	71.73 *	1800	311	58.73
40_3	5	46.17	3600	74.47 *	2152	315	61.3
40_4	5	60.14	3600	70.18 *	2213	176	16.69
40_5	5	49.63	3600	65.04 *	3963	489	31.05
50_1	8	63.58	3600	104.58 *	1684	378	64.49
50_2	8	71.44	3600	106.47 *	7200	520	49.03
50_3	8	80.99	3600	114.71 *	68	212	41.63
50_4	8	83.38	3600	119.11 *	17	459	42.85
50_5	8	69.34	3600	100.49 *	7200	427	44.92
Avg.		38.08	2770	50.94	1280		24.13

4.4 参数对成本的影响分析

为分析车辆数,车辆装载能力不同水平对成本是否有影响,针对单个算例 30\_1,考虑车辆数三种

水平( $nv=3,4,5$ ),车辆装载能力四种水平( $Q=7,8,9,10$ ),构造 12 个算例。对所有算例,采用改进分支切割算法,求得最优解,结果如下图所示。

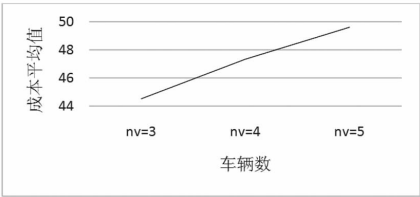


图 1 不同车辆数下平均成本折线图

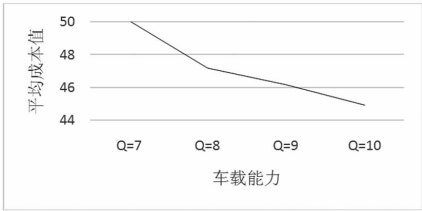


图 2 不同装载能力下平均成本折线图

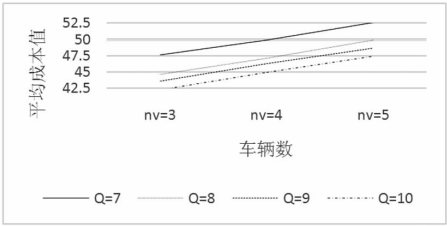


图 3 车载能力与车辆数交互作用下成本折线图

结果表明,(1)对于不同车辆数下的成本,则有成本随着车辆数的增多而增大(图 1 所示);(2)对于不同车载能力下的成本,则有成本随着车辆装载能力的增大而减少(图 2 所示)(3)对于车辆装载能力与车辆数交互作用下的成本,在不同的车辆数水平中,各个车辆装载能力下的成本按照相同的

规律变动,则有车辆装载能力和车辆数不存在交互作用(图 3 所示)。综上所述,车辆数和车辆装载能力不仅对成本有显著影响,而且车辆数越少,车辆装载能力越大,成本越少。

5 结论

本文研究了考虑订单取件时间和柔性时间窗的取送货车路线问题,并使配送成本和超时惩罚成本之和最小。首先,提出混合整数线性模型;根据该问题的特性,提出四种有效不等式;设计改进的分支切割算法。其次,通过数值算例,比较每种有效不等式的有效性,结果显示每种不等式能够不

同程度的提高模型下界;最后,在有限的运行时间内,与 CPLEX 的分支切割策略相比,验证算法的有效性。本文不仅可以丰富车辆路径问题文献,还为评价城市物流平台设计的启发调度算法性能提供参考。此外,适当的减少车辆数,适当的增加车辆装载能力,可有效的降低成本。

本文考虑了有确定性因素的带时间窗的取送货问题,而随着订单的需求个性化的不断增加,未来研究可以考虑订单分配和路径规划中的不确定因素。

## 参考文献:

- [1] 马艳芳,闫芳,康凯,李宗敏. 不确定同时取送货车辆路径问题及粒子群算法研究[J]. 运筹与管理,2018, 27(12):73-83.
- [2] 李寒梅. 多时间窗需求可拆分集送货车辆路径问题研究[D]. 华中科技大学,2016.
- [3] 祁文祥,陆志强,孙小明. 带软时间窗的集货与送货多车辆路径问题节约算法[J]. 交通运输工程学报, 2010,10(02):99-103.
- [4] 边展,张倩,徐奇,靳志宏. 带时间窗取送货问题的混合算法[J]. 运筹与管理,2020,29(02):97-107.
- [5] Sun P, Veelenturf L P, Hewitt M, et al. The time-dependent pickup and delivery problem with time windows[J]. Transportation Research Part B: Methodological, 2018, 116: 1-24.
- [6] Ghilas V, Cordeau J, Demir E, et al. Branch-and-price for the pickup and delivery problem with time windows and scheduled lines[J]. Transportation Science, 2018, 52(5): 1191-1210.
- [7] Ropke S, Cordeau J, Laporte G. Models and branch-and-cut algorithms for pickup and delivery problems with time windows[J]. Networks, 2007, 49(4): 258-272.
- [8] Furtado M G, Munari P, Morabito R. Pickup and delivery problem with time windows: a new compact two-index formulation[J]. Operations Research Letters, 2017, 45(4): 334-341.
- [9] Ropke S, Cordeau J. Branch and cut and price for the pickup and delivery problem with time windows[J]. Transportation Science, 2009, 43(3): 267-286.
- [10] Mahmoudi M, Zhou X. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: a dynamic programming approach based on state-space-time network representations[J]. Transportation Research Part B: Methodological, 2016, 89:19-42.
- [11] Baldacci R, Bartolini E, Mingozzi A. An exact algorithm for the pickup and delivery problem with time windows[J]. Operations Research, 2011, 59(2): 414-426.
- [12] Nanry W P, Barnes W J. Solving the pickup and delivery problem with time windows using reactive tabu search[J]. Transportation Research Part B: Methodological, 2000, 34(2): 107-121.
- [13] Goeke D. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles[J]. European Journal of Operational Research, 2019, 278(3): 821-836.
- [14] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows[J]. Transportation Science, 2006, 40(4): 455-472.
- [15] Li H, Lim A. A metaheuristic for the pickup and delivery problem with time windows[J]. International Journal on Artificial Intelligence Tools, 2003, 12(2): 173-186.