

基于实时信息的取送货动态车辆路径问题研究

孙宝凤, 史俊妍, 杨 雪, 郑再思

(吉林大学 交通学院, 吉林 长春 130022)

摘要: 为适应点对点、实时城市配送对动态响应和快速决策提出的新要求, 研究了多种实时信息作用和影响下的取送货动态车辆路径问题. 采用将动态问题转化为一系列静态问题的建模方法, 建立了基于实时信息的取送货动态车辆路径模型; 设计了动态算法框架, 运用构造算法获得初始可行解, 运用禁忌搜索算法改善初始可行解质量. 实验表明, 本文的模型和算法能有效解决基于实时信息的取送货动态车辆路径问题, 将初始可行解的质量(实时物流配送成本)改善了 34%.

关键词: 动态车辆路径规划; 取送货问题; 动态算法; 构造型算法; 禁忌搜索算法

中图分类号: U491

文献标志码: A

文章编号: 1001-5132 (2019) 03-0087-08

目前, 具有点对点、小批量、多批次特点的实时城市配送快速发展, 需求倍增, 对物流配送的及时响应和灵活性提出了新的技术要求. 考虑取送货的动态路径规划是提升配送服务高效性的技术途径之一, 简称 DVRP-PD 问题^[1].

目前基于实时信息的取送货动态车辆路径问题的研究比较少. 为了满足顾客对物流配送高效的要求, Ferrucci 等^[2]以惩罚成本最小为首要目标, 以车辆运营成本最小化为第二目标, 采用了禁忌搜索算法得出调度方案, 结果表明: 根据动态事件持续调整运输计划提高了调度方案的质量. 戈丽娜^[1]将取送货问题转化为整数线性规划模型, 目标函数是使总运行距离最小化, 未考虑车辆在行驶过程中与派遣中心的信息沟通和意外事故的情况. 杨茹^[3]在取送货动态车辆路径问题的研究上将单初始解的禁忌搜索算法改进为多初始解的禁忌搜索算法, 并将局部禁忌表扩展为全局禁忌表, 多初始解的禁忌算法可以节约 2.52% 的成本, 但同时也降低了运算效率. Jia 等^[4]提出了周期控制和事件驱动的滚动时域方法解决取送货动态车辆路径问题. Marjolein 等^[5]研究了考虑装卸货成本的取送货旅行商问题. Escuin 等^[6]针对带时间窗的动态取送货问题提出了系统等待策略. Lin 等^[7]提出了一定时间范围内的动态问题可以被视作一系列各个时刻

的静态问题的研究方法. 现有文献为 DVRP-PD 问题给出了研究框架, 即首先将动态问题分解为一系列静态子问题, 再逐一研究单一动态信息对决策的影响. 然而, 新请求增加信息研究较多^[4-5], 考察配送过程中全面动态信息对决策的影响分析却较为缺乏. 为此, 本文研究基于实时信息的 DVRP-PD 问题, 即在新请求逐渐出现、旧请求改变、交通堵塞、配送车辆抛锚这 4 种动态信息作用下的 DVRP-PD 问题. 首先界定问题, 建立基于实时信息的 DVRP-PD 模型; 然后, 针对模型设计动态求解算法, 给出算法步骤; 进而对已建立的模型和采用的算法进行试验与分析; 最后得到研究结论.

1 模型建立

1.1 问题描述

在实际生活中, 同城快递和网络外卖等在配送服务时, 每一个请求在下单后录入到配送服务商的系统中, 配送人员按照请求信息, 先到达取货点进行取货服务, 再将取到的货物运送到送货点, 由收货人签收. 在这一过程中, 每个请求有对配送服务时效性的要求, 而且这些请求在一天内不断地出现, 还存在已下单的请求要求修改配送信息的情况. 对于运输服务商, 需要考虑交通状况和车辆抛锚的意外事件, 在满足请求要求的同时降低

收稿日期: 2019-03-02.

宁波大学学报(理工版) 网址: <http://journal.g.nbu.edu.cn/>

基金项目: 国家自然科学基金(51308249); 博士学科点专项科研基金(20130061110008); 吉林省产业技术研究与开发专项(2015Y044).

第一作者: 孙宝凤(1970-), 女, 吉林长春人, 博导/教授, 主要研究方向: 物流系统仿真与供应链管理. E-mail: sunbf@jlu.edu.cn

配送成本.

因此本文研究的基于实时信息的 DVRP-PD 问题可定义如下: 考虑每一个请求的取货点和送货点必须满足前驱后继关系且只被访问一次, 同时满足配对关系, 即同一请求的取货点和送货点必须由同一辆车访问, 且不超过配送车辆的最大配送量和最大行驶距离, 考虑违反时间窗限制的附加成本和配送过程中实时变化的信息, 目标是使配送总成本最小.

除了新请求在调度过程中不断地出现并修改, 一些有动态本质的事件也会出现, 例如车辆拥堵造成的车辆减速和车辆抛锚等. 由于无法提前得到这些动态事件的信息, 只能实时处理这些问题. 在本文中即使车辆拥堵的持续时间是未知的, 顾客请求也不能由另一辆车进行转运. 但是, 如果发生了车辆抛锚的事件, 这辆车上的货物必须在抛锚的位置由另一辆车装载并完成运输.

车辆由第三方物流公司提供, 每辆车有一名司机, 每辆车的装载量、人工费用和以运行距离为基础的行驶成本都相同. 对于违反时间窗限制的司机(无论是由于交通拥堵还是车辆抛锚), 将按照超出时间窗的程度给以惩罚. 顾客对旧请求的改变要求只要是发生在车辆执行前就可以接受. 一辆车一旦从车场被派出, 就会不断寻求新的请求让这辆车完成. 找不到合适的请求时, 车辆就返回车场. 车辆运营成本与车辆数量和总行驶距离有关. 车辆总是选择两地间的最短路径行驶.

由于基于实时信息的 DVRP-PD 问题的时效性尤其重要, 因此, 目标函数应为最小化超出时间窗的惩罚成本和车辆运营成本的总和.

1.2 符号表示

(1)常量:

n 为客户请求的个数(车场用 0 表示);

P 为请求 $i \in \{1, \dots, n\}$ 的取货点, $P = \{1, \dots, n\}$;

D 为请求 $i \in \{1, \dots, n\}$ 的送货点, $D = \{n+1, \dots, 2n\}$;

q_i 为 i 节点的货物需求量, 当 $q_i > 0$ 时, q_i 表示取货点的货物需求量; q_{i+n} 表示相应送货点的货物需求量, $q_{i+n} = -q_i$;

$W_p(t)$ 为 t 时刻网络中所有车辆所在地的集合;

$|W_p(t)|$ 为 t 时刻网络中所有车辆所在地的个数;

$W_{p0}(t)$ 为 t 时刻网络中所有车辆所在地和车场的集合;

$W_u(t)$ 为 t 时刻车辆路径中尚未接受服务的客户请求(尚未接受服务的客户请求取消)和新增加的动态客户请求的集合, $W_u(t) \subseteq P \cup D$;

$W_{u0}(t)$ 为 t 时刻车辆路径中尚未接受服务的客户请求(尚未接受服务的客户请求取消)、新增加的动态客户请求和车场的集合, $W_{u0}(t) \subseteq N$;

$W_{up0}(t)$ 为 t 时刻所有车辆所在地、车辆路径中尚未接受服务的客户请求(尚未接受服务的客户请求取消)、新增加的动态客户请求和物流中心的集合;

T_i 为车辆完成一项请求(从物流中心出发至取货完毕回到物流中心, 成为一次取货任务)的最大行驶时间;

KP 为动态调整前车辆调度计划使用的车辆数;

K 为可使用的车辆总数;

T 为车辆完成一次任务(从车场出发至回到车场, 称为一次任务)的最大行驶时间;

$T_{rk}(t)$ 为 t 时刻车辆 k 从所在地出发时的已累积行驶时间(若车辆从车场出发, 则 $T_{rk}(t) = 0$), $r \in W_{p0}(t)$, $k \in \{1, 2, \dots, K\}$;

Q 为每辆车的额定车载量;

$Q_{ik}(t)$ 为 t 时刻车辆 k 从客户 i 出发时已累积的载重(若车辆从车场出发, 则 $Q_{ik}(t) = 0$), $k \in \{1, 2, \dots, K\}$;

d_i 为请求 i 的取货作业时间与送货作业时间, $i \in P \cup D$;

t_{ij} 为车辆由点 i 行驶到点 j 的时间, $i, j \in P \cup D$;

$[e_i, l_i]$ 为客户请求 i 的服务时间窗, 其中 e_i 为时间窗的开始时间, l_i 为时间窗的结束时间, $i \in \{1, \dots, n\}$;

α 为晚于时间窗 1 h 的惩罚费用;

β 为每台车辆的固定使用成本(包括车辆的折旧费、维修费等);

c 为单位路径成本.

(2)变量:

B_i 为车辆到达点 i 的时间, $i \in P \cup D$;

w_i 为车辆在取货点 i 的等待时间, $w_i = \max\{0, B_i - e_i\}$, $i \in P$;

td_i 为车辆在点 i 服务完后出发的时间, $i \in P$ 时, $td_i = B_i + w_i + d_i$, $i \in D$ 时, $td_i = B_i + d_i$;

tl_i 为请求 i 晚于时间窗(接受服务)的时间, $tl_i = \max\{td_i - l_i, 0\}$, $i \in D$;

z_k 为 0-1 变量, 如果生成初始车辆调度计划的车辆 k 被使用, 则 $z_k = 1$, 否则 $z_k = 0$, $k \in \{1, 2, \dots, K\}$;

n_x 为与动态调整前的车辆调度计划相比, 新使用的车辆数, $n_x = \max\{\sum_{k=1}^K z_k - KP, 0\}$, $k \in \{1, 2, \dots, K\}$.

(3) 决策变量:

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 由点 } i \text{ 驶向点 } j, \\ 0, & \text{其他,} \end{cases}$$

$$i \in W_{up0}(t), j \in W_{u0}(t), k \in \{1, 2, \dots, K\}.$$

1.3 模型建立

在一个特定时刻 t , 存在无向图 $G(N, A)$, 其中 $N = \{0, \dots, 2n\}$ 为节点集合, A 为弧集合. 对于每一个请求 $i \in \{1, \dots, n\}$, 在取货点 i 有起始时间窗限制, 当车辆在该时刻之前到达时, 车辆必须等待, 但不产生惩罚费用; 在送货点 $n+i$ 有结束时间窗限制, 当车辆在该时刻之后到达时, 产生惩罚费用. 为了加入同一请求, 相应取送货节点的顺序有前驱后继的约束和取送货车辆约束. 后者是指这一请求的取送货节点必须由同一辆车服务, 定义集合 S' , 任意 $S \subseteq S'$ 中, 至少有一个请求 i 的取货点满足 $i \notin S$ 且送货点 $n+i \in S$ [8].

本文提出基于实时信息的 DVRP-PD 模型为:

$$\min Z = \sum_{i \in W_{up0}(t)} \sum_{j \in W_{u0}(t)} c_{ij} \cdot x_{ijk} + \sum_{i \in W_{up0}(t)} \alpha \cdot tl_i + \beta \cdot n_x, \quad (1)$$

s.t.

$$\sum_{k=1}^K \sum_{i \in W_{up0}(t)} x_{ijk} = 1, \forall j \in W_{u0}(t), \quad (2)$$

$$\sum_{k=1}^K \sum_{i \in W_{u0}(t)} x_{ijk} = 1, \forall i \in W_{up0}(t), \quad (3)$$

$$x(S) \leq |S| - 2, \forall S \in S', \quad (4)$$

$$B_i + w_i + d_i + t_{ij} = B_j, i \neq j, \forall x_{ijk} = 1, \quad (5)$$

$$\sum_{i \in W_{u0}(t)} \sum_{j \in W_p(t)} \sum_{k=1}^K x_{ijk} = 0, \quad (6)$$

$$\sum_{i \in W_p(t)} \sum_{j \in W_{u0}(t)} \sum_{k=1}^K x_{ijk} = |W_p(t)|, \quad (7)$$

$$\sum_{i \in W_{u0}(t)} \sum_{j \in W_{u0}(t)} \sum_{k=1}^K x_{ijk} = \sum_{k=1}^K z_k, \quad (8)$$

$$Q_{ik} \geq \max\{0, q_i\} + \sum_{j \in W_{up0}(t)/\{i\}} \max\{0, q_j\} \cdot x_{ijk}, \quad \forall i \in W_{u0}(t), \quad (9)$$

$$Q_{ik} \leq \min\{Q_k, Q_k + q_i\} - (Q_k - \max_{j \in N/\{i\}}\{q_j\} - q_i)x_{0i} - \sum_{j \in W_{up0}(t)/\{i\}} \max\{0, q_j\} \cdot x_{ij}, \forall i \in W_{up0}(t), \quad (10)$$

$$\sum_{i \in W_{up0}(t)} \sum_{j \in W_{u0}(t)} x_{ijk} \cdot t_{ij} \leq T - T_{rk}, \forall k \in \{1, 2, \dots, K\}, \quad \forall r \in W_{up0}(t), \quad (11)$$

$$x_{ijk} \in \{0, 1\}, \forall j \in W_{up0}(t), \forall k \in \{1, 2, \dots, K\}, \quad (12)$$

$$n_x = \max\{\sum_{k=1}^K z_k - KP, 0\}, \forall k \in \{1, 2, \dots, K\}. \quad (13)$$

目标函数(1)表示 t 时刻动态调整后的车辆调度计划的总费用最少, 总费用包括车辆总行驶费用、晚于时间窗的惩罚费用和新使用车辆的固定使用成本费用 3 部分.

约束(2)和(3)表示每一个节点必须得到服务, 只能由一辆车完成; 约束(4)表示前驱后继约束, 表示对于请求 i , 取货点 i 必须在相应的送货点 $n+i$ 之前得到服务, 并且这对节点必须由同一辆车服务; 约束(5)表示车辆到达路径中相邻 2 节点时间之间的关系; 约束(6)和(7)表示对 t 时刻所有车辆所在地的访问是单向的, 只能发车, 然后回到车场; 约束(8)表示配送网络中所使用车辆的数量; 约束(9)和(10)表示容量约束; 约束(11)表示车辆 k 为完成承担的任务量而行驶的距离之和不大于车辆的剩余行驶距离; 约束(12)表示决策变量的取值范围; 约束(13)表示 t 时刻调整后动态车辆调度计划新使用的车辆数.

2 求解算法

2.1 动态算法

如图 1 所示, 动态算法或在线算法本质为动态环境中的算法框架. 动态算法将问题分割成许多静态子问题, 以方便构造型算法和改善型算法求得静态子问题的解.

图 1 中对调度时域子间隔进行求解部分, 采用启发式算法进行求解. 细分为 2 个阶段: 第 1 阶段采用构造型算法, 求出初始解; 第 2 阶段采用改善型算法(禁忌搜索算法), 改善初始解的质量.

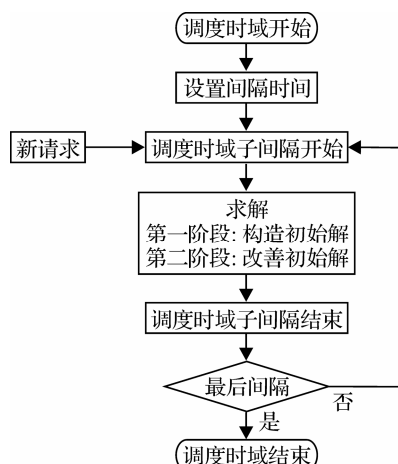


图1 动态算法流程

2.2 构造初始解

初始可行解一般采用构造型算法, 可以借鉴解决静态的带时间窗的取送货问题(Pickup and Delivery Problem with Time Windows, PDPTW)的算法, 例如贪心插入算法。贪心插入算法从一条空路径开始, 随机选择一个未安排的请求插入。对于每一个请求而言, 检查现存路径中所有可行的取货点和送货点插入位置。这里可行位置指满足取货点在送货点之前的约束、取货点与相应送货点在同一条路径上的约束、容量约束和时间窗约束。并且, 初始化一条新的路径插入该请求。比较所有可行位置, 选择适应度函数最小的插入位置。采用此方法将所有未被安排的请求插入到路径中。

适应度函数为:

$$\min Z = \sum_{i \in W_{up0}(t)} \sum_{j \in W_{d0}(t)} c_{ij} \cdot x_{ijk} + \sum_{i \in W_{up0}(t)} \alpha \cdot tl_i + \beta \cdot n_x$$

采用插入算法取得初始解, 步骤如下:

步骤 1 新建一条空路径;

步骤 2 设置 LocalMin 为无穷大;

步骤 3 从所有未进行规划安排的请求中随机选择一个请求, 如果没有需要安排的请求则结束算法;

步骤 4 从所有路径中选择一条路径, 如果没有路径则新建一条空路径插入该请求;

步骤 5 从步骤 4 中选择的路径中取货点的可行插入位置选择一个位置, 插入步骤 3 中请求的取货点, 如果没有取货点可行插入位置则跳转至步骤 4;

步骤 6 在步骤 5 的基础上相应送货点的可行

插入位置中选择一个位置, 插入步骤 3 中请求的送货点, 如果没有送货点可行插入位置则跳转至步骤 5;

步骤 7 计算新路径的适应度值 cost, 如果小于 LocalMin 则将 cost 的值赋给 LocalMin, 将这条路径赋给 r_best, 将这个请求赋给 request_best, 将取货点位置赋给 p_position, 将送货点位置赋给 d_position, 跳转至步骤 6;

步骤 8 将请求 request_best 的取货点和送货点分别插入到路径 r_best 的 p_position 位置和 d_position 位置, 如果路径为空路径则直接返回已经完成插入的路径, 否则消除步骤 1 中的空路径; 跳转至步骤 3。

2.3 改善初始解

采用禁忌搜索算法对初始解进行改善。

2.3.1 邻域移动

采用邻域移动获得候选解。如图 2 所示, 从初始解的一条路径中随机移除一对取货点和送货点, 再重新插入到初始解中的一条路径中。

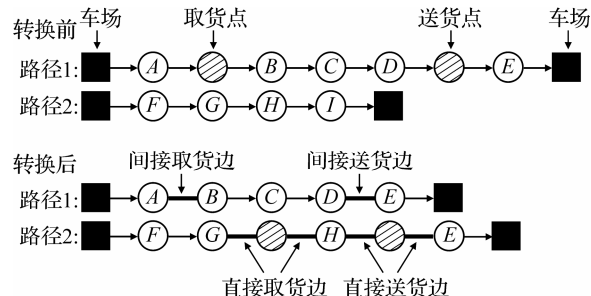


图2 邻域移动及禁忌对象

2.3.2 禁忌表

(1)禁忌对象。

禁忌对象为边, 如图 2 所示, 转换后的粗线边为禁忌对象。边 AB 为间接取货边, 边 DE 为间接送货边, 边 GH 为直接取货边, 边 HE 为直接送货边。所有引起这些边发生变化的邻域移动都是禁忌的。

(2)禁忌表长度和停止准则。

禁忌表长度和停止准则与静态子问题中需要重新规划的请求数量成正比。静态子问题中需要规划的请求数量在求解过程中是动态的, 需要在每一次运行禁忌搜索算法前实时确定。

(3)选择策略。

以候选解集为整个邻域, 从中选择一个适应

度函数为最小的解, 作为下一次迭代的初始解。

(4) 渴望水平。

当邻域中存在一个解的适应度函数值最小, 且比此次迭代的初始解的适应度函数值小, 则无论这个移动是否是禁忌的, 都接受这个解。

(5) 算法步骤:

步骤1 取得本次静态子问题中需要插入的请求数量, 并设置禁忌表长度和最大迭代次数;

步骤2 当迭代次数小于等于最大迭代次数时, 求得初始解的适应度函数值 $global_cost$, 设置候选解的个数为非禁忌的请求个数, 否则结束程序;

步骤3 随机选择非禁忌请求, 将其移除路径, 并记录间接取货边和间接送货边, 将移除的非禁忌请求重新插入到初始解中, 选择适应度函数值最小的插入位置, 并将新路径储存在候选解中, 直到没有可以选择的随机非禁忌请求;

步骤4 计算候选解的适应度函数值, 并按照从小到大的顺序进行排序;

步骤5 如果最小的适应度函数值小于 $global_cost$, 则直接接受这个候选解, 更新禁忌表, 否则, 接受不受到禁忌且适应度函数值最小的候选解, 并更新禁忌表;

步骤6 更新非禁忌请求, 更新迭代次数, 转至步骤2。

3 实验及分析

为了测试上述模型及算法的有效性, 随机生成一组随机点作为实验初始数据, 并进行路径规划和动态更新的测试。本文所有程序用 MATLAB 编写, 实验环境为 Intel(R) Core(TM) i7-4790 CPU 3.60 GHz。

3.1 实验数据

以 Li 等^[9]的数据为基础, 数据来源 <https://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark/>。

为了适应动态请求, 每个请求都有一个出现时间 $t_r = a \times t_r^{\text{latest}}$ 。其中, $t_r^{\text{latest}} = \min\{l_i, l_i - t_{ij} - s_i\} - t_{0i}$ 为请求最晚到达时间; a 为紧迫度, 取值范围为 $[0.1, 1]$, 步调为 0.1, a 越大, 紧迫度越高, 规划请求的时间越少, 也就是请求刚到便需要分配出去。

考虑车辆速度受到交通拥堵的影响, 在不同时段和不同位置有所不同, 按照规划静态子问题

的时间和车辆位置为不同位置的车辆配置不同的车辆速度, 即 v_{ijt} 表示在时段 T 内弧 ij 的速度。

为了清楚地说明本文的算法和模型在处理新请求逐渐出现、旧请求改变、车辆拥堵和车辆抛锚的实时信息时, 路径和调度规划的更新, 假定有 4 个请求, 为 $i(i=1, 2, 3, 4)$, i^+ 和 i^- 分别为取货点和送货点, 即 $(1^+, 2^+, 3^+, 4^+)$ 是取货点集合, $(1^-, 2^-, 3^-, 4^-)$ 是送货点集合。车辆的容量为 20, 起始点和返回点为车场。每一项请求的需求量、每个节点的位置坐标和每个节点的时间窗见表 1。

表1 实验初始数据

节点	需求量	位置	时间窗
车场	0	(15 000, 15 000)	
1^+	1	(17 033, 8 186)	[8:00, 9:00]
1^-	-1	(23 245, 16 818)	[8:00, 9:00]
2^+	4	(28 662, 11 573)	[8:01, 9:01]
2^-	-4	(21 071, 7 432)	[8:01, 9:01]
3^+	3	(6 819, 22 566)	[8:02, 9:02]
3^-	-3	(15 367, 17 305)	[8:02, 9:02]
4^+	5	(21 412, 9 298)	[8:40, 9:40]
4^-	-5	(28 644, 8 680)	[8:40, 9:40]

3.2 实验结果

请求 1, 2, 3 是在 8:05 时同时规划的, 可以视为静态请求, 动态请求见表 2。

表2 动态事件

序号	时间	事件
1	8:30	1^+ 的取货时间推迟至 8:50, 同时 1^- 推迟至 9:50 前送货
2	8:40	增加请求 4, 需求量为 5, 取货点为 4^+ , 送货点为 4^-
3	8:50	由于交通拥挤, 车 2 需要延迟 10 min
4	9:05	车 2 抛锚

在 8:05 时, 请求 1, 2, 3 和这 3 个请求的具体信息已知。路径规划的结果如图 3(a)所示, 配送总成本为 274.61 元。

在 8:30 时, 1^+ 的要求取货时间由 8:00 调整为 8:50, 即推迟了 50 min。因此, 如果按原计划车辆 2 不得不在节点 1^+ 等待至 8:50, 这样会使得车辆 2 无法在 9:00 前到达节点 1^- , 无法在 9:02 前到达节点 3^- , 从而产生高额的惩罚费用。通过重新规划路线, 得到图 3(b)的路线, 通过路线的调整, 使得路线上

产生的成本和惩罚成本相对降低, 为 252.37 元。

在 8:40 时, 增加了请求 4, 将需求量为 5 的货物由节点 4^+ 运送至节点 4^- 。通过更新路线, 如图 3(c)所示, 配送总成本为 241.39 元。

在 8:50 时, 车辆 2 遇到交通拥堵, 到达节点 1^+ 和节点 1^- 的时间往后顺延 10 min, 如图 3(d)所示, 配送总成本为 265.39 元。

在 9:05 时, 车辆 2 抛锚了, 重新规划路线时由车辆 1 先到达车辆 2 抛锚地点取货再送到相应的送货点, 车辆 2 由拖车送回停车场修理, 如图 3(e)所示, 配送总成本为 293.87 元。

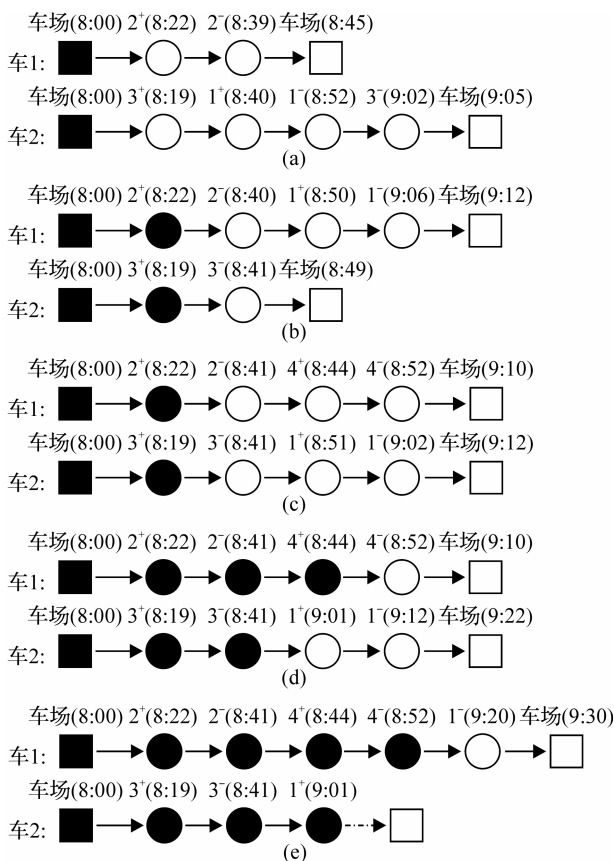


图3 路径规划结果

3.3 实验结果分析

3.3.1 全调度时域范围内解的质量分析

在规模为 53 个请求的动态取送货问题中, 比较调度时域的每一个间隔中构造解和“构造+禁忌”解的适应度函数的值, 即总成本。如图 4 所示, 纵坐标是调度时域每一个间隔适应度函数的值, 横坐标是调度时域的间隔数。由此可见, 经过改善的“构造+禁忌搜索”算法的结果总体上优于构造解

的结果, 解的质量比初始解大约提高了 34%。

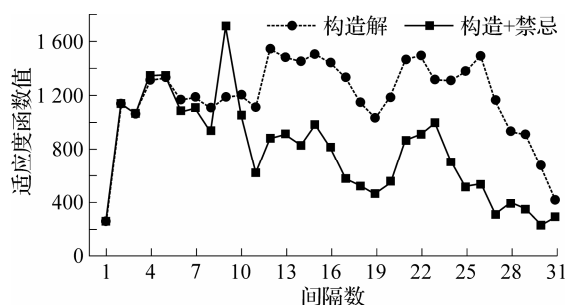


图4 构造解与“构造+禁忌”解的比较

3.3.2 不同调度时域间隔下解的质量比较

通过图 4 可以看出, 在整个调度时域过程中, 构造解与“构造+禁忌”的解在开始时相差不大, 中间相距较大。为了容易分辨, 在规划时域开始时, 路径数量相对较少, 分别取构造解和“构造+禁忌”解在间隔 1, 2, 4, 6 时的路径规划图进行对比, 分别如图 5~8 所示。图中“p”和“d”分别代表取货和送货节点; 红色实心圆代表车场, 红色空心圆圈代表车辆在间隔间的位置; 线条的颜色相同则代表同一辆车, 其中虚线代表已经行驶过的路径, 实线代表已规划好但还未行驶的路径。间隔 3 与间隔 2、间隔 5 与间隔 6 具有相似性, 故不做分析。

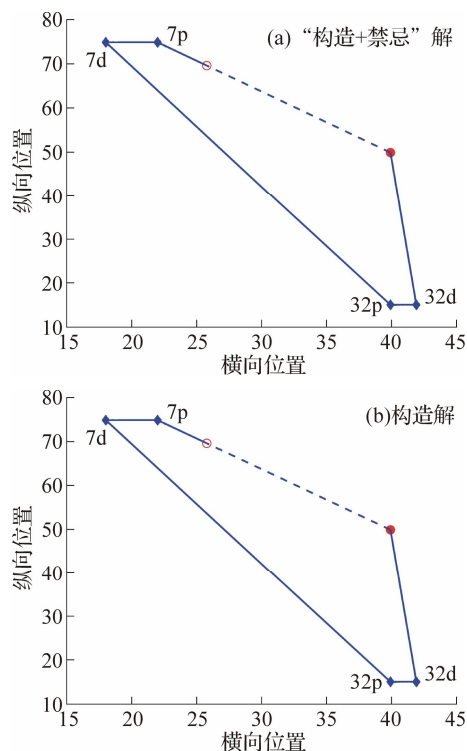


图5 间隔1的解

图5和图6分别代表间隔1和间隔2结束时的路径规划, 其中(a)代表“构造+禁忌”法得到的路径规划, (b)代表构造算法得到的路径规划. 可以看

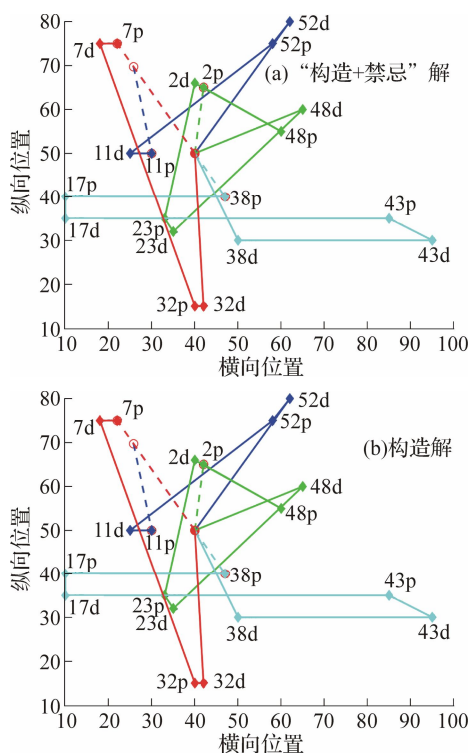


图6 间隔2的解

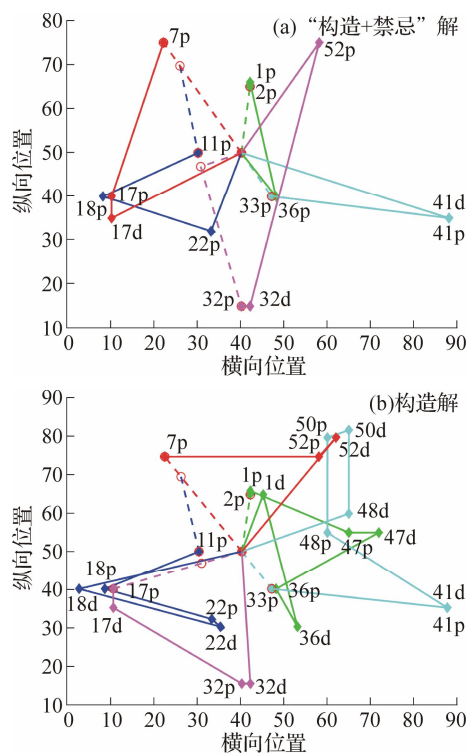


图7 间隔4的解

出, 在间隔1和间隔2时, 2种规划算法并没有区别, 二者在间隔1和间隔2的总成本都分别为263.09和1141.8元, 运算时间分别为0.03和0.48 s.

如图7所示, 间隔4“构造+禁忌”法的总成本开始优于构造法. “构造+禁忌”法得到的路径规划总成本为1315.9元, 运算时间为0.67 s; 构造算法得到的路径规划总成本为1346.6元, 运算时间为0.16 s. “构造+禁忌”法优化了路径的调度顺序, 且运算时间不超过1 s. 图8亦呈现出同样特征.

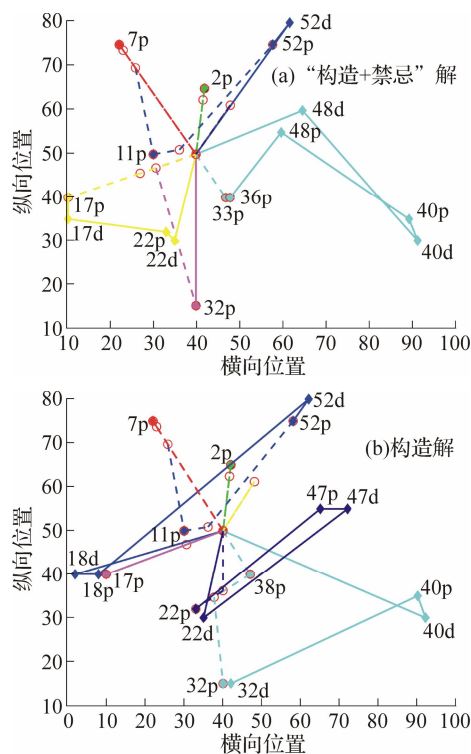


图8 间隔6的解

4 结论

本文采用已有的动态取送货问题的研究框架, 即将动态问题分解为一系列静态子问题, 综合考虑了新需求出现、需求改变、交通堵塞和配送车辆抛锚4种动态信息, 并建立了基于实时信息的取送货问题的模型; 将已有的静态数据改编为适用于动态问题的数据, 采用动态算法进行求解, 并进行结果分析, 得出的主要结论如下:

(1) 本文的模型和算法能够有效地解决基于实时信息的取送货动态车辆路径问题. 设计了动态算法框架, 将构造算法与禁忌搜索算法相结合, 运用构造算法获得初始可行解, 运用禁忌搜索算法

改善了初始可行解的质量, 改善幅度约为 34%.

(2)算例中动态请求规模为 53, 模型及其算法适用于点对点、小批量、多批次的实时配送, 可以降低现实中实时配送的成本, 并改善配送服务的质量.

(3)全调度时域范围为 1~31 个间隔, 算例表明, “构造+禁忌搜索”算法总成本曲线低于构造法总成本曲线, 因此经过改善的“构造+禁忌搜索”算法解的质量总体上优于构造解的结果, 且每次运行的计算时间不超过 1 s.

(4)不同调度时域间隔下, 解(总成本)的质量是变化的. 其中, 当未处理节点数较少时, 如间隔 1 和 2 情形, 解是相同的, 体现不出 2 种规划算法的差异. 当节点数增多时, 如间隔 3 至 6, 乃至 31 情形时, 2 种规划算法的区别较大. 禁忌搜索算法优化路径中节点的调度顺序, 从而改善了解的质量.

参考文献:

- [1] 戈丽娜. 配送过程中提货送货问题的静态动态方法的应用效果研究[D]. 哈尔滨: 哈尔滨工业大学, 2016.
- [2] Ferrucci F, Bock S. Real-time control of express pickup and delivery processes in a dynamic environment[J]. *Transportation Research Part B: Methodological*, 2014, 63:1-14.
- [3] 杨茹. 集送货一体化的动态车辆调度问题研究[D]. 天津: 南开大学, 2007.
- [4] Jia Y J, Wang C J, Wang L M. A rolling horizon procedure for dynamic pickup and delivery problem with time windows[C]. *IEEE International Conference on Automation and Logistics*, Shenyang, 2009:2087-2091.
- [5] Marjolein V, Kees J R, Iris F V, et al. The pickup and delivery traveling salesman problem with handling costs [J]. *European Journal of Operational Research*, 2017, 257(10):118-132.
- [6] Escuin D, Larrodé E, Millán C. A cooperative waiting strategy based on elliptical areas for the dynamic pickup and delivery problem with time windows[J]. *Journal of Advanced Transportation*, 2016, 50(8):1577-1597.
- [7] Lin C H, Choy K L, Ho G T S, et al. A decision support system for optimizing dynamic courier routing operations [J]. *Expert Systems with Applications*, 2014, 41(15): 6917-6933.
- [8] 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.
- [9] Li H, Lim A. A metaheuristic for the pickup and delivery problem with time windows[C]. *IEEE International Conference on Tools with Artificial Intelligence*, 2001:160-167.

Solution for dynamic pickup and delivery problem based on real-time information

SUN Bao-feng, SHI Jun-yan, YANG Xue, ZHENG Zai-si

(College of Transportation, Jilin University, Changchun 130022, China)

Abstract: To meet the updated requirements for dynamic response and quick decision-making in the field of point-to-point real-time urban distribution, the dynamic vehicle routing problem with pick-up and delivery is studied using the real-time information. The dynamic problem is transformed into a series of static problems, and a dynamic vehicle routing model with pickup and delivery based on real-time information is established; The dynamic algorithm framework is designed, and the initial feasible solution is obtained using the construction algorithm, and the quality of the initial feasible solution is improved by using the tabu search algorithm. The experimental results show that the proposed model and algorithm can effectively solve the dynamic vehicle routing problem with pickup and delivery based on real-time information. The quality of the initial feasible solution, the cost of real-time logistics distribution, are both improved by 34%.

Key words: vehicle routing; pickup and delivery problem; dynamic algorithm; constructive algorithm; tabu search algorithm

(责任编辑 韩 超)