

文章编号: 1001-0920(2012)04-0535-07

需求可拆分车辆路径问题的聚类求解算法

刘旺盛^{1,2}, 杨帆¹, 李茂青¹, 陈培芝¹

(1. 厦门大学信息科学与技术学院, 福建 厦门 361005; 2. 集美大学现代物流研究中心, 福建 厦门 361021)

摘要: 针对传统的车辆路径问题通常假设客户的需求不能拆分, 即客户的需求由一辆车满足, 而实际上通过需求的拆分可使需要的车辆数更少, 从而降低配送成本的问题, 分析了需求可拆分的车辆路径问题的解的特征, 证明了客户需求不宜拆分应满足的条件, 设计了符合解的特征的聚类算法, 并对其求解. 通过实验仿真, 将所提出的聚类算法与蚁群算法和禁忌搜索算法进行比较, 所得结果表明了所提出的算法可以更有效地求得需求可拆分车辆路径问题的优化解, 是解决需求可拆分车辆路径问题的有效方法.

关键词: 需求可拆分车辆路径问题; 聚类算法; 启发式算法

中图分类号: TP14

文献标识码: A

Clustering algorithm for split delivery vehicle routing problem

LIU Wang-sheng^{1,2}, YANG Fan¹, LI Mao-qing¹, CHEN Pei-zhi¹

(1. School of Information Science and Technology, Xiamen University, Xiamen 361005, China; 2. Modern Logistics Research Center, Jimei University, Xiamen 361021, China. Correspondent: LI Mao-qing, E-mail: mqli@xmu.edu.cn)

Abstract: In the traditional vehicle routing problems, customer demands are usually assumed that they can not be split. That is to say, a customer can only be served by a vehicle. In fact, split delivery requires fewer vehicles, which reduces transportation costs. This paper analyzes the solution's characteristics of split delivery vehicle routing problem, and also proves the situations that customers' demands can not be split. A clustering algorithm meeting the solution's characteristics is designed to solve this problem. Compared with ant colony algorithm and tabu search algorithm, the proposed algorithm is demonstrated to obtain optimal solution more effectively through the simulation, and it is an effective method to solve split delivery vehicle routing problem.

Key words: split delivery vehicle routing problem; clustering algorithm; heuristic algorithm

1 引言

车辆路径问题 (VRP) 自 1959 年由 Dantzig 等人^[1]提出以来, 一直是运筹学、组合优化领域研究的前沿与热点问题. 但是, 目前大部分关于车辆路径问题的研究^[2-3]大都预先设定了 1 个条件, 即每个客户的需求 (指小于车辆最大运载能力的需求) 必须由 1 辆车, 1 次服务中完成. 然而, 对于实际中的物流运作, 有时 1 个客户所要求的货物较多, 超出了车辆承载能力, 这时必须对客户的需求进行拆分, 而且在能满足服务要求的前提下, 有时通过需求的拆分可以更好地降低运输成本, 尤其在大部分客户需求量略大于 1/2 车辆运载能力的情况下 (Archetti 等人^[4]已经对此通过大量的实验进行了验证). 该问题由 Dror 等

人^[5]于 1989 年提出, 称为需求可拆分的车辆路径问题 (SDVRP), 目前已经成为车辆路径问题中一个较新的分支. 文献 [5] 证明了在客户需求可分的情况下对客户进行分割运输, 无论是总运输距离, 还是派车数目都可以进一步优化 VRP, 尽管该问题依然具有 NP-hard 属性. 从检索到的文献看, 求解 SDVRP 的算法主要集中在精确算法和启发式算法上, 算法需要解决的一个关键问题是如何对需求进行合理优化的拆分. Frizzell 等人^[7-8]讨论了以网格距离划分运输网络条件下的 SDVRP 求解算法, 以及时间窗对问题和求解过程的影响, 从而说明了时间窗对于算法的寻优能力有一定影响; Lee 等人^[9]将其视为动态规划问题求解; Jin 等人^[10]提出了带有效不等式的两阶段法;

收稿日期: 2010-10-25; 修回日期: 2011-04-11.

基金项目: 福建省自然科学基金项目(2010J01359); 国家“211”(三期)项目; 潘金龙基金项目(ZC2011002).

作者简介: 刘旺盛(1979—), 女, 博士生, 从事车辆路径优化、物流系统建模及仿真等研究; 李茂青(1953—), 男, 教授, 博士生导师, 从事系统工程、决策理论等研究.

Feillet 等人^[11]提出了有时间窗 SDVRP 问题的集合覆盖形式, 并采用列生成法求解. 但是, 精确算法求解问题的规模有限, 只能解决规模很小的问题, 因此近似算法也是解决该类问题的重要手段. 近似算法主要有 Dror 等人^[6]提出的局部搜索算法, Archetti 等人^[12]提出的禁忌搜索算法, 以及 Archetti 等人^[13]在禁忌搜索求解的基础上提出的进一步的优化算法.

国内研究主要着重于近似算法的设计上. 隋露斯等人^[14]设计了蚁群算法; 谢毅^[15]在其硕士论文中设计了禁忌搜索算法和遗传算法, 并对 2 种算法进行了对比, 发现遗传算法求解的整体质量和速度不如禁忌搜索算法. 总体而言, 相关的文献和成果较少, 现有的近似算法也均带有随机搜索性质, 且假定每个客户的需求均小于车辆最大运载能力. 为此, 本文考虑客户的需求可以超过车辆最大运载能力, 证明在此情况下应先单独满足, 剩余部分再联合配送, 然后采用先分组后路径的求解方法. 分组阶段拟运用 K -means 算法的思想, 结合需求可拆分车辆路径问题的特征设计合适的聚类算法, 将客户分组, 每组由一辆车负责配送. 每组内的线路优化即是 TSP 问题, 可采用模拟退火算法求解.

2 SDVRP 的数学模型

本文研究的是单车场、单车型、无时间窗要求、纯装货(或纯卸货)的需求可拆分的车辆路径问题. 设 C 为客户点集合, $C = 1, 2, \dots, n$, 集合 $C_0 = 0$ 代表车场; d_{ij} 为集合 C 和 C_0 内任意 2 点 i, j 之间的距离; q_i ($i = 1, 2, \dots, n$) 为客户 i 的需求量; w 为车辆最大运载能力; R 为完成任务所需的车辆数, 其取值为

$$R = \left\lceil \sum_{i=1}^n \frac{q_i}{w} \right\rceil, \quad (1)$$

“ $\lceil \cdot \rceil$ ”表示向上取整数, 如 $\lceil 6.2 \rceil = 7$; x_{ij}^r ($r = 1, 2, \dots, R$, $i, j = 0, 1, \dots, n$ 且 $i \neq j$) 为决策变量, 当且仅当第 r 条路线中车辆通过弧 (i, j) 时, $x_{ij}^r = 1$, 否则 $x_{ij}^r = 0$; y_i^r ($r = 1, 2, \dots, R, i = 1, 2, \dots, n$) 为第 r 条线路中满足客户 i 的需求量; S^r 代表第 r 条路线中服务的客户集合, $|S^r|$ 表示集合 S 所包含的元素个数. 模型假设:

- 1) 2 点间的距离是对称的, 即 $d_{ij} = d_{ji}$;
- 2) 点与点之间的距离关系符合三角形不等式, 即 $d_{ik} + d_{kj} > d_{ij}$;
- 3) 所有的车辆必须从车场出发, 完成一趟任务后, 必须返回到车场;
- 4) 每个客户的需求必须满足, 但可以由 1 辆或 1 辆以上的车辆来满足.

该问题的目标是合理安排车辆路线, 使得车辆运行成本尽可能少. 在本问题中, 车辆运行总成本用路

线总长度表示. 根据以上描述, 可得到该问题的数学模型如下:

$$\min \sum_{r=1}^R \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}^r. \quad (2)$$

s.t.

$$\sum_{i=0}^n x_{ik}^r = \sum_{j=0}^n x_{kj}^r, \quad k = 0, 1, \dots, n, \quad r = 1, 2, \dots, R; \quad (3)$$

$$\sum_{r=1}^R \sum_{i=0}^n x_{ij}^r \geq 1, \quad j = 0, 1, \dots, n; \quad (4)$$

$$\sum_{r=1}^R y_i^r = q_i, \quad i = 1, 2, \dots, n; \quad (5)$$

$$\sum_{i \in S^r} \sum_{j \in S^r} x_{ij}^r = |S^r| - 1, \quad r = 1, 2, \dots, R, \quad (6)$$

$$S^r \subseteq C - C_0;$$

$$\sum_{i=1}^n y_i^r \leq w, \quad r = 1, 2, \dots, R; \quad (7)$$

$$\sum_{j=0}^n x_{ij}^r q_j \geq y_{ri}, \quad r = 1, 2, \dots, R, \quad i = 1, 2, \dots, n; \quad (8)$$

$$x_{ij}^r \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad r = 1, 2, \dots, R; \quad (9)$$

$$q_i \geq y_i^r \geq 0, \quad i = 1, 2, \dots, n, \quad r = 1, 2, \dots, R. \quad (10)$$

约束 (2) 为行驶总路程最小化的目标; 约束 (3) 表示流量守恒, 即进入某点的车辆数与离开该点的车辆数相等; 约束 (4) 和 (5) 确保每个点至少被访问一次且需求均得到满足; 约束 (6) 表示每条线路中被服务客户之间的弧边数等于被服务客户点的个数减 1; 约束 (7) 为车辆运载能力限制; 约束 (8) 表示当且仅当车辆路过客户 i 时, 该客户才能得到服务.

3 解的基本特征

定理 1 如果 1 个 SDVRP 存在可行解, 且点与点之间的距离关系符合三角形不等式, 则当客户需求等于车辆最大运载能力, 即 $q_i = w$ 时, 该问题优化中的该点应单独满足, 不宜拆分.

证明 设某客户点 B 的需求量等于车辆运载能力 w , 该点的需求被拆分, 分别由线路 r_k 和 r_{k+1} 满足, 各线路中被满足的需求分别记为 y_{kB} 和 $y_{(k+1)B}$. 将线路 r_k 和 r_{k+1} 的其他需求点聚合为 1 点, 分别记为 A 和 C , 此 2 点的最大需求量分别为 $w - y_{kB}$ 和 $w - y_{(k+1)B}$. 车场用 O 点表示, B 的坐标位置相对于 AC 连线有 3 种情况: 1) 在 AC 连线的右侧; 2) 在 AC 连线的左侧; 3) 在 AC 连线上.

若 B 点需求单独满足, 则此时 A, C 两点的需求之和 q_{AC} 最大应为

$$\text{Max } q_{AC} = (w - y_{kB}) + (w - y_{(k+1)B}) = w,$$

因此A, C两点可以合并到1条线路中, 则被拆分满足和单独满足的路线分别如图1~图3所示。

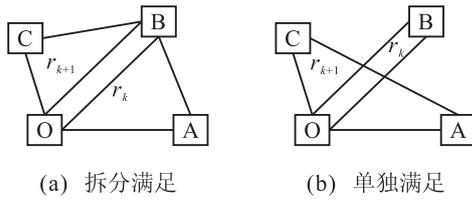


图1 B在AC连线的右侧

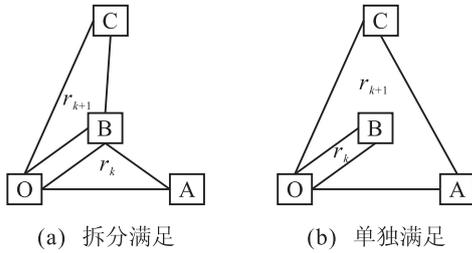


图2 B在AC连线的左侧

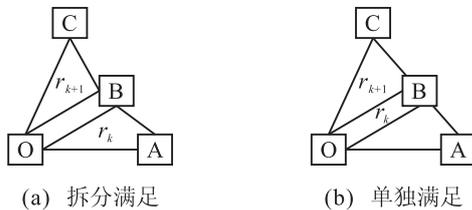


图3 B在AC连线上

由图1~图3可见, 3种情况下拆分满足的总路线长度 $Z_{\text{split}} = d_{OC} + d_{CB} + 2d_{OB} + d_{BA} + d_{AO}$, 单独满足的总路线长度 $Z_{\text{unsplit}} = d_{OC} + d_{CA} + d_{AO} + 2d_{OB}$, 则 $Z_{\text{split}} - Z_{\text{unsplit}} = d_{CB} + d_{BA} - d_{CA}$.

在情况1)和情况2)下, 由三角形2边之和大于第3边可知 $d_{CB} + d_{BA} - d_{CA} > 0$; 在情况3)下, $d_{CB} + d_{BA} - d_{CA} = 0$. 因此, 当客户需求量等于车辆最大运载能力时, 适宜单独满足, 不宜拆分. □

定理2 如果1个SDVRP存在可行解, 且点与点之间的距离关系符合三角形不等式, 则在该问题优化的任意2条路线中最多只会存在1个共同点.

由定理2可以得出以下2个推论:

推论1 如果1个SDVRP存在可行解, 且点与点之间的距离关系符合三角形不等式, 则在该问题优化中不会存在 k -split cycle.

k -split cycle是指对于 k 个客户可以构造 k 条线路, 线路 r_m 包含客户 i_m 和 i_{m+1} ($m = 1, 2, \dots, k-1$), r_k 包含客户 i_1 和 i_k , 此种构造称作 k -split cycle. 图4是1个3-split cycle.

推论2 如果1个SDVRP存在可行解, 且点与点之间的距离关系符合三角形不等式, 则在该问题优化中被拆分需求点的数量小于路线的数量.

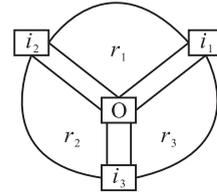


图4 3-split cycle

定理2以及推论1和推论2的证明请参见文献[5]和[16], 此略.

4 求解SDVRP问题的聚类算法

求解SDVRP问题的聚类算法是一个启发式算法, 它将问题的求解分为2个阶段, 第1阶段对客户进行聚类, 确定由同一辆车服务的客户集合; 第2阶段确定每个聚类的行走路线, 即TSP问题的求解. 常见TSP问题的求解算法有最近邻点法、最近插入法、树算法、蚁群算法、遗传算法及模拟退火算法等, 算法相对比较成熟, 其中的模拟退火算法具有计算过程简单、鲁棒性强、运行效率高和较少受到初始条件约束等优点, 因此本文采用模拟退火算法求解TSP问题, 将探讨的重点放在聚类算法上.

4.1 k-means 算法

许多实际应用中, 基于划分聚类的 k -means 聚类算法(又称 k -均值算法)已经被证明是高效的, 而且能够产生令人满意的聚类结果. 一般 k -means 算法的计算步骤如下:

Step 1: 选择 k 个初始聚类中心;

Step 2: 逐个将需分类的样本 x 按最小距离原则分配给某个聚类中心;

Step 3: 计算各个聚类中心新的向量值, 一般以各聚类域中所包含样本的均值向量为新的聚类中心, 需要分别计算 k 个聚类中的样本均值向量;

Step 4: 将样本逐个重新分类, 重复迭代计算, 直至各聚类中心不再移动时, 算法收敛, 计算完毕.

k -means 算法尝试找出使平方误差函数值最小的 k 个划分, 试图使生成的结果簇尽可能地紧凑和独立. 对于大数据集, 该算法是相对可伸缩和高效的, 因为它的复杂度是 $O(nkt)$. 其中: n 为所有对象的数目, k 为簇的数目, t 为迭代的次数.

4.2 聚类原则和拆分点选择原则

将 k -means 算法的思想运用于车辆路径问题的客户聚类中, 目的是使距离近的客户聚成一类, 由1辆车服务, 降低问题求解的复杂度. 但是, 由于车辆运载能力的限制, 它属于带约束条件的聚类, 而 k -means 算法针对的是不带任何约束条件的聚类问题, 因而不能将其聚类原则直接运用于车辆路径问题的求解. 在对车辆路径问题中的客户点进行聚类时, 需

考虑每个聚类客户点的需求总量不能超过最大运载量, 当某一类客户需求总量达到车辆最大运载量时, 其他客户便不得不选择次近的聚类加入. 因此, 车辆路径问题的聚类原则不能仅遵循距离最近准则, 还应遵循次近准则. 另外, 若客户优先加入准则不同, 则聚类结果也不同. 本文采取距离优先准则, 即当客户 i 和 j 均距离聚类中心 P 最近时比较 d_{ip} 与 d_{jp} 的大小, 选择距离较小者先加入. 具体聚类过程如下:

Step 1: 随机确定 R (由式 (1) 获得) 个聚类中心.

Step 2: 计算各客户点到各聚类中心的距离 d_{ij} ($i = 1, 2, \dots, n, j = 1, 2, \dots, R$), 将客户点 i ($i = 1, 2, \dots, n$) 到各聚类中心的距离 d_{ij} ($j = 1, 2, \dots, R$) 从小到大排序, 得到各客户点到聚类中心的最小距离.

Step 3: 将各客户点到聚类中心的最小距离从小到大排序, 若最小值为 d_{kp} , 则将该最小值对应的客户 k 加入聚类 p , 再将次小值对应的客户加入其对应的聚类, 并计算聚类剩余需求量 SuQ (即车载量减去已装载量), 依次类推, 当聚类剩余需求量小于欲加入客户的需求量时, 在聚类中选择拆分点进行拆分 (拆分点选择原则稍后探讨).

Step 4: 当客户欲加入的聚类的总需求量已达到车辆最大运载量时, 将考虑次近的聚类, 依此类推, 直至所有客户均加入某个聚类中.

在需求可拆分的车辆路径问题中, 为保证车辆的满载率, 使需要的车辆数最少, 允许客户的需求拆分满足, 因此需考虑拆分客户选择原则. 当客户 i 加入一未满载聚类 p , 使聚类的需求总量超过车辆最大运载量时, 需求需要拆分满足. 但是, 若该客户的次近聚类中心距离客户较远时, 将大大增加车辆行驶距离. 因此, 允许未满足需求传导到聚类 p 中距离其他聚类 (剩余需求量必须大于客户 i 未满足的需求量) 较近的点 (需求量必须大于客户 i 的未满足需求量), 将该点需求拆分满足, 客户 i 的需求则全部由聚类 p 满足. 若所有聚类的剩余需求量均小于客户 i 的未满足需求量, 则选择剩余需求量最大的优先加入, 这样可以避免被拆分次数过多.

4.3 求解步骤

本文探讨的是客户需求有可能大于车辆最大运载量的情况. 因此, 由定理 1, 对于客户需求大于或等于车辆最大运载能力的情况, 宜优先整车满足, 而余下的部分可与其他客户点合并满足, 即作为需要聚类的客户点.

下面运用第 4.2 节讨论的原则对需要聚类的客户点进行聚类, 确定由同一辆车服务的客户. 但是, 因为需求可拆分的车辆路径问题是带约束条件的聚类, 因

此计算不一定收敛, 需要设置迭代次数 N 强行终止, 并设置聚类评估准则选出迭代中较好的聚类结果. 本文采用的聚类评估函数为

$$\text{Min}(\text{Sum } D) = \sum_{j=1}^R \sum_{i \in C_j} d_{ij}. \quad (11)$$

式中 C_j 代表聚类 j , 即计算聚类中的每个客户点到该类中心的距离之和, 选取距离之和最小的作为最优聚类结果. 算法具体步骤如下:

Step 1: 寻找需求量大于或等于车辆运载能力的客户点, 将其需求量 q_i 拆分为 q_i^s 和 q_i^c 两部分, 其中

$$q_i^s = w \lfloor q_i/w \rfloor, \\ q_i^c = q_i - w \lfloor q_i/w \rfloor,$$

式中 $\lfloor \cdot \rfloor$ 表示向下取整数, 如 $\lfloor 6.6 \rfloor = 6$. 该点需求量 q_i^s 单独满足, 余下需求量 q_i^c 与其他点合并到某一回路中满足, 将该点需求量修改为 q_i^c .

Step 2: 从客户集合 $C = 1, 2, \dots, n$ 中任意选取 R 个初始的聚类中心 $1^1, 2^1, \dots, R^1$, 记集合 $P^1 = 1^1, 2^1, \dots, R^1$, 初始化各聚类集合 $C_i = \emptyset, i = 1, 2, \dots, R$, 设定最大迭代次数 N 的值.

Step 3: 客户聚类. 计算各客户点到各聚类中心的距离 d_{ij} , 找出各客户对应的最近聚类中心, 距离近的优先加入. 若欲加入的聚类已满载, 则选择次近的聚类加入; 当欲加入的聚类还有剩余需求量, 但加上客户 i 的需求量聚类 C_j 的总需求量 $QC_j > W$ 时, 则计算客户 i 未满足的需求量, 记为 S , 并将未能满足的需求量传导到聚类 C_j 中其他客户上. 传导原则如下: 先找出聚类 C_j 中所有需求量不小于 S 的客户点 (包括客户 i), 再找出聚类剩余需求量 $SuQ_z = W - QC_z \geq S$ ($z \in P^1 - j$) 的聚类, 计算这些客户点到剩余需求量大于 S 的聚类的距离, 选择距离最小的 1 个客户点进行拆分. 若该客户点为 k , 对应的聚类中心为 p , 则将客户 k 加入聚类 C_p , 即未满足的需求量 S 由该线路满足; 若所有聚类的剩余需求量 $SuQ_z < S$, 则选择剩余需求量最多的聚类加入, 直至未满足需求量 S 全部得到满足; 依此类推, 直至所有客户需求得到满足.

Step 4: 计算各聚类客户点到其聚类中心的距离之和 $\text{Sum } D$.

Step 5: 用如下方式调整类的中心, 得到新的类中心为 $1^2, 2^2, \dots, R^2$. 聚类中心 j^2 ($j = 1, 2, \dots, R$) 的坐标分别为

$$x_{j^2} = \frac{1}{n_i} \sum_{x_i \in C_i} x_i, \quad y_{j^2} = \frac{1}{n_i} \sum_{y_i \in C_i} y_i,$$

式中 n_i 是 C_i 中所包含的客户点的数目.

Step 6: 重复 Step 3 ~ Step 5, 直至最大迭代次数 N , 输出 $\text{Sum } D$ 值最小时对应的聚类结果.

Step 7: 采用模拟退火法优化 Step 6 得出的最优聚类方案中各聚类的线路. 降温方式为

$$T(t+1) = k \times T(t).$$

式中: k 为正的略小于 1.00 的常数, t 为降温的次数.

该算法 Step 1 考虑了客户需求大于车辆运载能力的情况, Step 2~Step 6 将需要聚类的客户聚类, 并寻找最优聚类方案, Step 7 进行线路优化, 得到 TSP 问题的解.

5 实验比较分析

为测试算法的有效性, 本文采用以往文献中的数据进行测试, 并与现有的蚁群算法、禁忌搜索算法等求解效果较好的启发式算法进行比较. 实验中, 聚类后的路线优化采用模拟退火算法求解. 因为分组后每组客户数较少, 得到最优解需要迭代的次数少, 所以设定模拟退火初始温度 $t_0 = 10$, 取 $k = 0.87$. 算法由 Matlab 7.1 实现, 在 CPU 为 3.40 GHz, 内存为 512 M 的 Pentium(R)D 计算机上运行.

实验 1 文献 [14] 设计了蚁群优化算法, 用以求解 15 个客户点的 SDVRP 问题. 15 个客户点的坐标和需求量如表 1 所示, 配送中心的坐标为原点, 车辆的最大运载量为 500.

表 1 15 个客户点的基本信息

客户点	横坐标	纵坐标	需求量
1	32	41	468
2	96	9	335
3	7	58	1
4	97	87	170
5	26	21	225
6	23	100	479
7	52	31	359
8	76	43	463
9	74	17	465
10	72	104	206
11	40	99	146
12	8	16	282
13	27	38	328
14	78	69	462
15	46	16	492

本文取聚类迭代次数 $N = 500$, 距离采用欧几里得距离. 利用所提出的算法进行测试, 10 次实验结果如表 2 所示. 由表 2 可知, 该算法求解性能稳定, 10 次实验求得回路数均为 10, 最优路线总长度为 1764.4, 比文献 [14] 的蚁群算法得到的最优路线总长度 1836 减少 3.9%, 而总距离长度平均值 1800.17 也比其降低了 1.95% 左右. 然而, 文献 [14] 没有给出蚁群算法的求解速度, 所以在求解速度方面难以作出比较. 但是, 由表 2 可知, 该算法的求解速度较快, 一次求解只需 5~6s. 实验中最优总距离值对应的路径如表 3 所示, 括号内数字表示该点在该回路中的需求满足量. 显

然, 该最优解中任意 2 条路线最多只存在 1 个共同点, 且不存在 k -split cycle, 符合定理 2 和推论 1. 被拆分的客户点为 9, 11, 12, 14 和 15, 被拆分的客户数小于路线 10, 因此也符合推论 2. 经实验验证, 该算法求得的解均符合前述解的基本特征, 且该算法的求解效能比文献 [14] 设计的蚁群算法高.

表 2 本文算法求解文献 [14] 算例的计算结果

计算次数	路线总长度	运算时间/s
1	1867.6	5.450
2	1764.8	5.292
3	1801.5	5.142
4	1799.9	5.332
5	1793.0	5.290
6	1801.9	5.280
7	1872.1	5.349
8	1764.4	5.158
9	1771.7	5.258
10	1764.8	5.207
平均值	1800.17	5.276

表 3 本文算法求得文献 [13] 的最优路径

回路编号	回路路径(各回路中各点需求满足量)
1	0-5(225)-12(275)
2	0-6(479)-11(21)
3	0-8(463)-9(37)
4	0-1(468)-3(1)
5	0-9(263)-15(237)
6	0-4(170)-14(330)
7	0-12(7)-13(328)-15(165)
8	0-11(125)-10(206)-14(132)
9	0-2(335)-9(165)
10	0-7(359)-15(90)

实验 2 文献 [15] 设计了禁忌搜索算法和遗传算法, 用以求解 20 个客户的 SDVRP 问题, 并将禁忌搜索算法与遗传算法进行了比较, 发现遗传算法求解的整体质量和速度不如禁忌搜索算法, 因此本文不再与遗传算法进行对比分析. 文献 [15] 的禁忌搜索算法计算结果如表 4 所示.

表 4 文献 [15] 的禁忌搜索算法计算结果

计算次数	路线总长度	运算时间/s	最佳解迭代步数
1	193.1	163	537
2	189.2	175	805
3	197.9	170	966
4	186.4	166	881
5	195.0	169	946
平均值	192.32	168.6	827

同样取迭代次数 $N = 500$, 距离采用欧几里得距离. 利用本文算法测试, 10 次实验结果如表 5 所示. 由表 4 和表 5 可知, 本文算法比禁忌搜索算法所求得的结果优, 路线总长度平均值减少 5% 左右, 最优路线总长度降低 3.73%, 且运算时间大大低于禁忌搜索算法所需要的运算时间, 经过较少的迭代次数便能找到较优解. 当改变 N 的取值, 令其等于 100 时, 也能得到较

好的结果. 经过 6 次实验, 得到的总路线长度平均值为 185.683, 运算时间平均为 4.191 s, 各测试指标均优于文献 [15] 中的禁忌搜索算法.

表 5 本文算法求解文献 [15] 算例的计算结果

计算次数	路线总长度	运算时间/s
1	184.49	5.515
2	182.12	5.170
3	184.57	4.911
4	179.45	5.069
5	184.74	5.010
6	181.12	5.053
7	184.49	5.586
8	182.12	5.174
9	184.57	4.938
10	179.45	5.068
平均值	182.712	5.1494

实验 3 文献 [12] 也设计了禁忌搜索算法来求解 SDVRP 问题. 算法由 C++ 编程实现, 在内存为 256 MB, CPU 为 2.40 GHz 的奔腾 4 计算机上运行. 实验结果表明, [12] 设计的禁忌搜索算法优于 [6] 中的局部搜索算法, 因此本文不再与局部搜索算法进行比较. 所测试的 42 个例子中每 7 个为一组, 第 1 组为原问题, 后缀为“00”, 来自于 [16] 的第 1-5, 10 和 11 个问题, 故分别用“P01-00, …, P05-00, P10-00 和 P11-00”表示, 需服务的客户数分别为 50, 75, 100, 150, 199, 199 和 120, 第 2~第 6 组为原问题的变异, 对客户点的需求量进行了变动, 通过后缀体现, 如“P01-1030”表示客户点需求量均介于车辆最大运载量的 10%~30%, 其他特征均未改变. 利用本文算法求解该 42 个算例, 实验结果比较见表 6. 表 6 中本文算法聚类迭代的次数为 500 次, 路径总长度为计算 10 次得到的最好值, 平均运算时间为 10 次运算时间的平均, 改进百分比的计算公式为

$$\frac{\text{禁忌搜索求得路径总长} - \text{本文算法求得路径总长}}{\text{禁忌搜索求得的路径总长}}$$

负数表示没有改进. 由表 6 可以看出, 大部分结果好于禁忌搜索求得的解, 而且计算 1 次所需要的平均运算时间远远小于禁忌搜索算法, 需要的最长时间为 98.0 s, 而禁忌搜索所需要的最长时间为 11 347 s.

6 结 论

在车辆路径优化问题中, 允许将客户的需求拆分满足, 这有利于节约车辆资源, 缩短最优路径. 本文分析了需求可拆分情况下车辆路径问题的解的特征, 证明了当客户需求等于车载容量时不宜拆分, 应单独配送. 根据解的特征, 设计了符合解的特征的聚类算法. 实验表明, 该算法性能较好, 可以较快地找到问题的优化解或近似优化解, 是解决需求可拆分车辆路径问题的有效方法. 但是, 算法仍有进一步改进的空间, 比如在初始聚类中心的选取、聚类准则及聚类评估函数

表 6 本文算法求解文献 [12] 算例的计算结果比较

算例	本文算法		禁忌搜索算法		改进
	路径总长	平均运算时间/s	路径总长	平均运算时间/s	
P01-00	4 675 585	4.6	5 307 907	17	11.9
P02-00	8 158 990	7.9	8 542 757	64	4.5
P03-00	8 149 102	9.5	8 413 577	60	3.1
P04-00	10 696 819	15.2	10 708 613	440	0.1
P05-00	13 682 582	22.5	13 403 505	1 900	-2.1
P10-00	13 929 231	22.3	13 403 505	40	-3.9
P11-00	9 972 833	10.7	10 569 587	86	5.6
P01-1030	4 228 238	5.6	4 629 056	27	8.7
P02-1030	5 966 489	9.2	6 239 394	78	4.4
P03-1030	7 618 932	13.6	7 714 649	122	1.2
P04-1030	9 915 250	24.3	9 471 386	545	-4.7
P05-1030	12 895 562	36.1	11 482 700	1 224	-12.3
P10-1030	12 575 805	35.8	11 482 700	516	-9.5
P11-1030	12 074 965	17.9	10 552 825	85	-14.4
P01-1050	6 894 370	7.0	7 653 121	56	9.9
P02-1050	9 809 075	11.7	11 340 760	71	13.5
P03-1050	14 331 431	17.6	15 151 732	206	5.4
P04-1050	21 142 701	31.8	21 018 042	564	-0.6
P05-1050	27 019 924	47.4	25 858 494	3 811	-4.5
P10-1050	27 011 949	46.2	25 858 494	259	-4.5
P11-1050	32 827 882	23.8	30 604 668	188	-7.3
P01-1090	9 747 314	9.5	10 391 059	34	6.2
P02-1090	15 100 849	16.6	15 566 936	311	3.0
P03-1090	19 555 495	25.2	20 541 296	412	4.8
P04-1090	31 193 493	48.7	29 916 416	1 822	-4.3
P05-1090	41 245 476	72.4	36 242 004	2 598	-13.8
P10-1090	41 002 814	70.6	36 242 004	1 037	-13.1
P11-1090	50 395 453	35.7	45 026 152	523	-11.9
P01-3070	13 413 544	9.4	15 119 826	52	11.3
P02-3070	20 900 718	16.3	23 386 654	184	10.6
P03-3070	29 907 109	25.1	31 552 228	454	5.2
P04-3070	45 644 618	47.7	46 741 320	1 512	2.3
P05-3070	55 677 116	73.5	57 158 484	2 279	2.6
P10-3070	55 628 673	71.8	57 158 484	477	2.7
P11-3070	63 266 507	34.0	73 501 136	411	13.9
P01-7090	14 726 636	12.4	15 039 466	160	2.1
P02-7090	22 370 386	21.6	22 935 488	437	2.5
P03-7090	32 090 091	33.6	30 709 048	1 891	-4.5
P04-7090	49 302 168	62.2	44 968 584	8 783	-9.6
P05-7090	61 484 472	98.0	55 711 292	11 347	-10.4
P10-7090	61 146 681	95.6	55 711 292	2 033	-9.8
P11-7090	78 230 554	52.2	71 682 608	1 865	-9.1

等方面尚有待进一步探讨. 另一方面, 可以引入时间窗和多车型等在实际问题中经常出现的限制条件进一步研究.

参考文献(References)

- [1] Dantzing G, Ramser J. The truck dispatching problem[J]. *Management Science*, 1959, 10(6): 80-91.
- [2] 赵燕伟, 彭典军, 张景玲, 等. 有能力约束车辆路径问题的量子进化算法[J]. *系统工程理论与实践*, 2009, 29(2): 159-166.
(Zhao Y W, Peng D J, Zhang J L, et al. Quantum

- evolutionary algorithm for capacitated vehicle routing problem[J]. *Systems Engineering-Theory & Practice*, 2009, 29(2): 159-166.)
- [3] 李相勇, 田澎. 开放式车辆路径问题的蚁群优化算法[J]. *系统工程理论与实践*, 2008, 28(6): 81-93.
(Li X Y, Tian P. Research on ant colony optimization algorithm for open vehicle routing problem[J]. *Systems Engineering-Theory & Practice*, 2008, 28(6): 81-93.)
- [4] Claudia Archetti, Martin W P, Savelsbergh M, et al. To split or not to split: That is the question[J]. *Transportation Research-Part E*, 2008, 44(1): 114-123.
- [5] Dror M, Trudeau P. Savings by split delivery routing[J]. *Transportation Science*, 1989, 23(2): 141-145.
- [6] Dror M, Trudeau P. Split delivery routing[J]. *Naval Res Logist*, 1990, 37(3): 383-402.
- [7] Frizzell P, Giffin J. The bounded split delivery vehicle routing problem with grid network distances[J]. *Asia Pacific J of Operational Research*, 1992, 9(1): 101-116.
- [8] Frizzell P W, Gi In J W. The split delivery vehicle scheduling problem with time windows and grid network distances[J]. *Computers & Operational Research*, 1995, 22(6): 655-667.
- [9] Lee C G, Epelman M A, White C C, et al. A shortest path approach to the multiple-vehicle routing problem with split picks-ups[J]. *Transportation Research B*, 2006, 40(4): 265-284.
- [10] Jin M, Liu K, Bowden R O. A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem[J]. *Int J of Production Economics*, 2007, 105(1): 228-242.
- [11] Dominique Feillet, Pierre Dejax, Michel Gendreau, et al. Vehicle routing with time windows and split deliveries[R]. Avignon: Laboratoire Informatique d'Avignon, 2006.
- [12] Archetti C, Hertz A, Speranza M G. A tabu search algorithm for the split delivery vehicle routing problem[J]. *Transportation Science*, 2006, 40(1): 64-73.
- [13] Archetti C, Savelsbergh M W P, Speranza M G. An optimization-based heuristic for the split delivery vehicle routing problem[J]. *Transportation Science*, 2008, 42(1): 22-31.
- [14] 隋露斯, 唐加福, 潘震东, 等. 用蚁群算法求解需求可拆分车辆路径问题[C]. 中国控制与决策会议. 烟台, 2008: 997-1001.
(Sui L S, Tang J F, Pan Z D, et al. Ant colony optimization algorithm to solve split delivery vehicle routing problem[C]. *Chinese Control and Decision Conf. Yantai*, 2008: 997-1001.)
- [15] 谢毅. 需求可拆分的物流车辆路线问题研究[D]. 上海: 同济大学经济与管理学院, 2006.
(Xie Y. The study of logistics vehicle routing problem with split demand[D]. shanghai: School of Economics & Management, Tongji University, 2006.)
- [16] Gendreau M, Hertz A, Laporte G. A tabu search heuristic for the vehicle routing problem[J]. *Management Science*, 1994, 40(10): 1276-1290.

(上接第534页)

- [5] Yan H S, Xu D. An approach to estimating product design time based on fuzzy ν -support vector machine[J]. *IEEE Trans on Neural Networks*, 2007, 18(3): 721-731.
- [6] Hao P Y. New support vector algorithms with parametric insensitive/margin model[J]. *Neural Networks*, 2010, 23(1): 60-73.
- [7] Hong D, Hang C. Interval regression analysis using quadratic loss support vector machine[J]. *IEEE Trans on Fuzzy Systems*, 2005, 13(2): 229-237.
- [8] Tanaka H, Uejima S, Asai K. Linear regression analysis with fuzzy model[J]. *IEEE Trans on Systems, Man and Cybernet*, 1982, 12(6): 903-907.
- [9] Hong W C. Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model[J]. *Energy Conversion and Management*, 2009, 50(1): 105-117.
- [10] 徐海龙, 王晓丹, 廖勇, 等. 一种基于 PSO 的 RBF-SVM 模型优化新方法[J]. *控制与决策*, 2010, 25(3): 367-370.
(Xu H L, Wang X D, Liao Y, et al. New approach for optimizing model of RBF-SVM based on PSO[J]. *Control and Decision*, 2010, 25(3): 367-370.)
- [11] Wu Q. Regression application based on fuzzy ν -support vector machine in symmetric triangular fuzzy space[J]. *Expert Systems with Applications*, 2010, 37(4): 2808-2814.
- [12] 许多. 产品开发时间分析与估计方法的研究[D]. 南京: 东南大学自动化学院, 2004.
(Xu D. The analysis and estimation method for product development time[D]. Nanjing: School of Automation, Southeast University, 2004.)