

## 两阶段启发式算法求解带时间窗的多中心车辆路径问题

于 滨, 靳鹏欢, 杨忠振

(大连海事大学 交通运输管理学院, 大连 116026)

**摘 要** 车辆路径问题 (VRP) 是物流研究领域中的一个具有重要理论价值和现实意义的问题。带时间窗的多中心车辆路径优化问题 (MDVRPTW) 是单中心带时间窗的 VRP (VRPTW) 的一个扩展, 其非常复杂, 难于求解。本文提出一个两阶段的启发式算法来求解 MDVRPTW。该算法首先通过基于聚集度的启发式分类算法将 MDVRPTW 简化为多个 VRPTW; 然后采用蚁群算法对每个 VRPTW 进行求解。为了提高蚁群算法的效率, 提出了两个改进策略: 交叉算子和自适应的 ant-weight 信息素增量更新策略。最后, 通过若干经典的 MDVRPTW 对该算法进行了验证, 结果显示结合基于聚集度的启发式分类算法和改进的蚁群算法是一个求解 MDVRPTW 的有力工具。

**关键词** 带时间窗的多中心车辆路径问题; 聚集度; 分类算法; 蚁群算法

## Two-stage heuristic algorithm for multi-depot vehicle routing problem with time windows

YU Bin, JIN Peng-huan, YANG Zhong-zhen

(Transportation Management College, Dalian Maritime University, Dalian 116026, China)

**Abstract** Vehicle routing problem (VRP) plays a vital role in logistics research. The multi-depot vehicle routing problem with time windows (MDVRPTW), an extension of VRP with time windows (VRPTW), is very complicated to be solved. This paper presented a two-stage heuristic to solve MDVRPTW. In the two-stage heuristic, aggregation-based clustering algorithm was firstly proposed to transfer MDVRPTW to several VRPTWs, and ant colony optimization (ACO) was developed to optimize each VRPTW. Two improvement strategies: crossover operation and adaptive ant-weight strategy, are designed to improve ACO. The performance of the two-stage heuristic was examined by use of some classic instances. Results show that the two-stage heuristic is a powerful tool for MDVRPTW.

**Keywords** multi-depot vehicle routing problem with time window; aggregation; clustering algorithm; ant colony optimization

### 1 引言

带时间窗的车辆路径问题 (VRPTW) 是物流研究领域中的一个具有重要理论价值和现实意义的问题。由于车辆路径问题通常被看作是一个内嵌旅行商求解 (TSP) 的问题, 因此, 其被认为是 NP-hard 问题。带时间窗多中心车辆路径问题 (MDVRPTW) 可以看成是 VRPTW 的泛化, 其求解相对于单中心问题更加困难。国外在 MDVRPTW 方面的研究主要有: Polacek 等<sup>[1]</sup> 提出一种基于变邻域搜索方法来求解 MDVRPTW; Cordeau 等<sup>[2-3]</sup> 提出了一种禁忌算法以及若干改进措施来求解 MDVRPTW 等问题; 其他的多中心车辆路径问题 (MDVRP) 的研究还包括文献 [4-9]。国内在 MDVRPTW 方面的研究还比较少, 目前部分学者对 MDVRP 进行了探讨, 主要包括: 王平等<sup>[10]</sup> 建立了多源点物流配送方案优化模型。李臻等<sup>[11]</sup> 提出针对车场满载的多车场车辆调度的优化模型, 并设计基于贪婪算法的启发式算法进行求解。李敏等<sup>[12]</sup> 借鉴 Floyd 算法研究了多中心的车辆配送问题。另外, 还有部分学者对带时间窗的车辆路径问题进行了研究, 主要包括:

**收稿日期:** 2010-06-08

**资助项目:** 国家自然科学基金青年基金 (51108053); 教育部人文社会科学研究青年基金项目 (10YJC630357); 中国博士后特别资助 (201003611); 大连海事大学基本业务费项目 (2011ZC029, 2011QN037)

**作者简介:** 于滨 (1977-), 男, 副教授, E-mail: minlfish@yahoo.com.cn; 杨忠振 (1964-), 男, 教授, E-mail: yangzhongzhen@263.net.

李相勇等<sup>[13]</sup>基于禁忌算法求解带时间窗和随机时间的车辆路径问题. 丁秋雷等<sup>[14]</sup>分析了使用蚁群算法求解带时间窗的车辆路径问题的可行性等.

对于车辆路径问题, 随着配送点数量的增加, 精确算法的计算时间增长很快<sup>[15]</sup>. 启发式算法被认为是求解复杂、大规模的组合优化问题的有力工具, 而且大量的研究也表明, 采用启发式算法是求解车辆路径问题的有效途径. 因此, 本文提出一个两阶段启发式算法: 基于聚集度的启发式分类算法和改进的蚁群算法来求解 MDVRPTW. 首先通过启发式分类算法对客户进行分类, 然后通过改进的蚁群算法针对各客户群体进行单中心的路径优化.

## 2 带时间窗的多中心车辆路径问题

带时间窗的多中心车辆路径问题是研究车辆从多个中心出发对若干客户进行服务的问题. 在配送过程中, 要求每辆车只能从一个配送中心出发, 完成配送工作后必须返回同一配送中心; 这个过程需要在规定的时间内服务所有客户, 同时每位客户只能被服务一次; 车辆在配送过程中的载货量不能超过车容量. 典型的多中心车辆路径问题可以描述如下(见图 1):  $C = \{c_1, c_2, \dots, c_H\}$  表示所有客户的集合;  $D = \{d_1, d_2, \dots, d_N\}$  表示配送中心的集合; 每个客户 ( $c_i \in C$ ) 相应的需求为  $q_i (q_i > 0)$ ;  $L$  表示所有边的集合.  $[T_{c_h}^S, T_{c_h}^E]$  为客户  $c_h$  的访问时间窗约束条件, 也就是车辆服务该客户的时间最早不能早于  $T_{c_h}^S$ , 最迟不能晚于  $T_{c_h}^E$ .

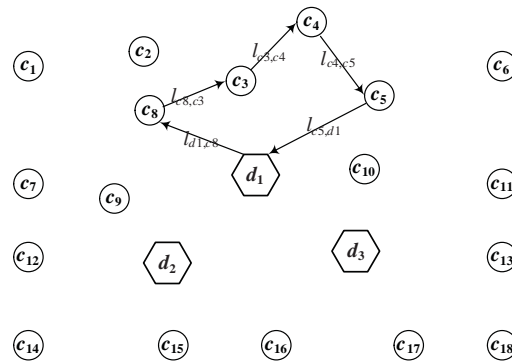


图 1 典型的带时间窗的多中心车辆路径问题

## 3 两阶段启发式算法

本文提出的两阶段启发式算法是采用的“先分类, 后优化”的思想. 1) 首先, 基于配送中心将所有的客户进行分类, 即把多中心优化问题简化为单中心优化问题. 该方法可以大幅降低计算问题的复杂性, 提高算法搜索效率. 然而, 客户分类的准确与否直接关系到最后的优化质量, 因此, 仅采用距离配送中心的远近来划分客户非常容易降低优化质量. 因此, 本文参考蚁群算法的转移规则, 引入一个类似信息素的多维启发式信息来进行客户的分类. 2) 然后, 本文通过一个改进的蚁群算法对多个单中心的 VRPTW 进行优化. 具体的算法如下所述.

### 3.1 基于聚集度的启发式分类算法

比较常用的客户分类方法是基于客户与配送中心距离来进行分类, 其优点是比较简单, 缺点是分类的准确程度较差. 特别是针对带有时间窗的问题, 这种分类方法不十分可靠. 另外, 传统的分类方法也很少考虑先前分类的经验, 从而导致分类算法效率不高. 在进行客户分类时, 主要有两类信息需要考虑, 一类是确定性信息, 包括客户相对于配送中心的空间分布状况以及客户之间的衔接性(时间窗是否相似). 另一类信息是启发式信息, 这类信息与分类的准确程度成正比, 也即是, 根据每次分类结果所得到的方案对该类信息进行调整, 并对后续搜索进行指导. 这样, 参考蚁群算法的转移规则, 对于客户  $c_h$  选择配送中心  $d_n$  的概率  $p(c_h, d_n)$ , 就可以表示为:

$$p(c_h, d_n) = \frac{\omega(c_h, d_n)^\mu \times \psi(c_h, d_n)^\gamma}{\sum_{d_n \in D} \omega(c_h, d_n)^\mu \times \psi(c_h, d_n)^\gamma} \quad (1)$$

式中,  $\omega(c_h, d_n)$  表示分类过程中的启发式信息, 其与分类的准确程度相关, 这里, 将其称为分类信息素;  $\psi(c_h, d_n)^\gamma$

表示分类过程中的确定性信息, 这里, 将其称为聚集度;  $\mu, \gamma$  是控制参数, 用来控制两类信息的相对重要程度.

### 1) 分类信息素计算

分类信息素与蚁群算法的信息素相似, 其与优化方案的质量相关, 这里用优化方案中所有配送线路的总长度的倒数来计算. 该信息随着搜索的进行, 会不断更新, 为后续搜索提供引导信息, 本文采用了传统的、比较简单的更新公式:

$$\omega^{new}(c_h, d_n) = \rho \times \omega^{old}(c_h, d_n) + \frac{N \times Q}{\sum_{d_n \in D} F_{d_n}}, \quad \rho \in (0, 1) \quad (2)$$

式中,  $\omega^{old}(c_h, d_n), \omega^{new}(c_h, d_n)$  分别表示客户  $c_h$  与配送中心  $d_n$  之间更新前后的分类信息素的数量;  $\rho$  为常数, 表示分类信息素的蒸发速度.  $N$  表示配送中心数量;  $Q$  为常量;  $F_{d_n}$  表示配送中心  $d_n$  的解决方案的目标函数值 (所有路径的长度).

### 2) 聚集度信息计算

聚集度与客户 - 配送中心间的距离和客户之间时间窗相似程度有关. 这里, 聚集度采用下式来计算:

$$\psi(c_h, d_n) = \frac{1}{l_{c_h, d_n}} \times \delta(c_h, d_n) \quad (3)$$

式中,  $l_{c_h, d_n}$  表示客户  $c_h$  与配送中心  $d_n$  的距离,  $\frac{1}{l_{c_h, d_n}}$  被用来表示客户相对于配送中心的分布状况;  $\delta(c_h, d_n)$  被用来表示客户  $c_h$  与当前配送中心  $d_n$  的当前的客户集的时间衔接性. 其可以通过客户间的运行时间和客户的时间窗来计算:

$$\delta(c_h, d_n) = \frac{\sum_{c_{h'} \in C_{d_n}} \sigma(c_h, c_{h'})}{M_{d_n}} \quad (4)$$

式中,  $C_{d_n}$  表示配送中心  $d_n$  当前的客户集合;  $M_{d_n}$  表示配送中心  $d_n$  当前的客户集合中客户的数量;  $\sigma(c_h, c_{h'})$  可以按下式计算:

$$\sigma(c_h, c_{h'}) = \begin{cases} 1, & \text{如果 } (T_{c_h}^S + t_{c_h, c_{h'}} < T_{c_h}^E) \text{ 或者 } (T_{c_{h'}}^S + t_{c_{h'}, c_h} < T_{c_{h'}}^E) \\ 0, & \text{其他} \end{cases} \quad (5)$$

式中,  $T_{c_h}^S, T_{c_h}^E$  分别表示客户  $c_h$  时间窗的开始和结束时刻;  $t_{c_h, c_{h'}}$  表示客户  $c_h$  与客户  $c_{h'}$  之间的运行时间.

很明显, 由于存在时间衔接性参数, 当前的客户集会影后续客户的配送中心的选择. 为了减小这种影响, 本文采用的是一种随机的客户选择方法, 即随机选择一个客户, 然后再应用公式 (1) 进行配送中心选择. 图 2 显示了一个客户分类的例子. 例如, 客户  $c_9$  虽然离配送中心  $d_2$  更近, 但是由于时间衔接性的关系, 其被划分到配送中心  $d_1$ .

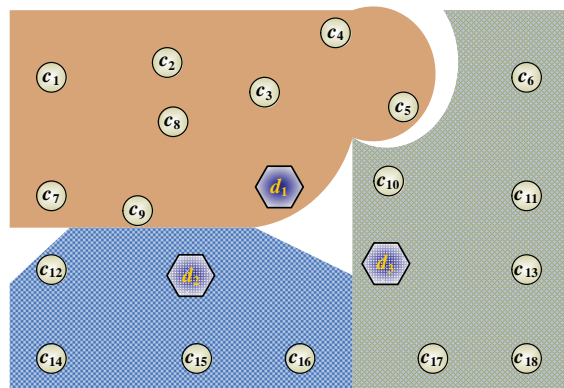


图 2 一个基于聚集度启发式分类算法的实例

## 3.2 改进的蚁群算法

蚁群算法是受自然界中蚁群觅食行为启发而提出的一种模拟进化算法, 其是由意大利学者 Dorigo<sup>[16]</sup> 等人在 20 世纪 90 年代初首先提出来的. 蚁群算法作为一种新的启发式算法已经成功地应用到了许多领域, 在车辆路径问题方面也有很多学者进行了大量的研究<sup>[16-18]</sup>. 本文通过客户分类算法将多中心的车辆路径问题划分为若干个单中心车辆路径问题, 然后通过蚁群算法对其进行分别求解.

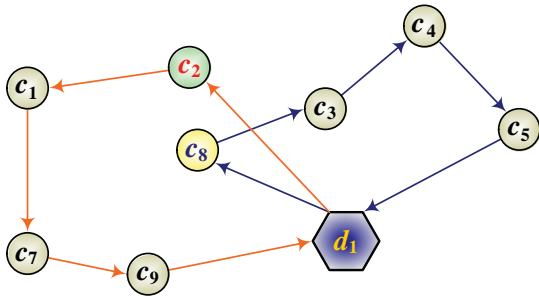


图 3 选择交叉点

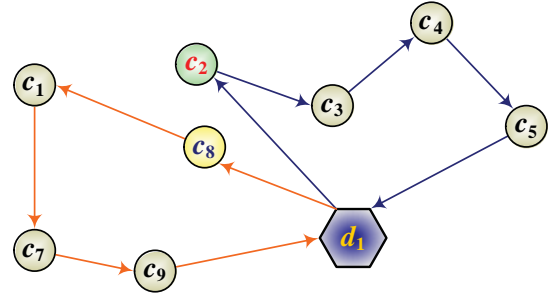


图 4 交换两个交叉点

### 1) 方案构造

在本文提出的蚁群算法中, 一个成功解决方案由若干从中心出发, 最后返回该中心的路线组成. 这些线路必须能够为所有客户提供服务, 同时每位客户只能被服务一次, 而且必须满足每个客户的时间要求和车容量的限制等约束. 线路的编码采用的是基于车辆行驶路径的顺序编码形式, 例如, 图 1 中的一条从中心  $d_1$  出发的线路的编码为:  $d_1 - c_8 - c_3 - c_4 - c_5 - d_1$ .

### 2) 转移规则

蚂蚁从中心出发, 通过逐步选择服务的客户来构造线路. 首先, 蚂蚁会选择其在当前位置的可行客户集, 然后, 基于一定的规则从可行客户中选择下一个客户. 这个规则通常与各条路段上的信息素浓度和期望值有关, 为了区分进行分类时的信息素, 这里采用路径信息素来代替信息素. 这样, 对于第  $i$  点来说, 选择  $j$  的概率如下:

$$p_{ij} = \frac{(\tau_{ij})^\alpha \times (\eta_{ij})^\beta}{\sum_{h \notin tabu} (\tau_{ih})^\alpha \times (\eta_{ih})^\beta} \quad (6)$$

式中,  $\tau_{ij}$  表示路段  $(i, j)$  上路径信息素的数量;  $\eta_{ij}$  表示路段  $(i, j)$  上的期望值;  $\alpha$  和  $\beta$  分别表示路径信息素和能见度的启发式因子;  $tabu$  表示不可行的节点的集合, 例如, 线路已经经过的点以及不满足时间约束的点就是不可行点.

### 3) 交叉操作

在蚁群算法中引入交叉算子可以有效的抑制算法陷入局部收敛, 提高优化质量<sup>[17]</sup>. 本文采用了一个简单的交叉操作, 具体的交叉过程如下:

**Step 1** 从解决方案中随机选择两条路径, 然后再从两条路径分别随机选择一个客户作为交叉对象, 例如, 第一条路径客户  $c_8$  和第二条路径客户  $c_2$  被选择, 如图 3.

**Step 2** 交换客户  $c_8$  和客户  $c_2$  构造新的解决方案, 如图 4.

**Step 3** 由于客户是随机选择的, 所以很容易产生劣解, 因此有必要使用一种局部搜索算法来改进解决方案的局部最优性. 本文采用了 2-opt 法<sup>[17-18]</sup> 来进行局部优化.

与遗传算法相似, 这里也引入一个交叉率 ( $p_m$ ) 来控制每个方案是否需要交叉. 交叉率不宜过大, 否则, 会影响算法的收敛速度. 通常, 在优化初期应该在尽可能大的搜索空间内优化, 随时间推移优化越来越精细, 不宜出现过大的跳跃以阻碍收敛. 因此, 本文采用了一种自适应的方法确定交叉概率, 具体的公式如下:

$$p_m(t) = p_m^{\min} + (p_m^{\max} - p_m^{\min})^{1-t/T} \quad (7)$$

式中,  $p_m^{\min}$  和  $p_m^{\max}$  分别表示最小和最大的交叉率, 两个概率分别为,  $p_m^{\min} = 1/H$  ( $H$ : 表示网络中客户数量) 和  $p_m^{\max} = 1/N$  ( $N$ : 表示配送中心个数);  $T$  和  $t$  分别表示最大的进化代和当前的进化代.

### 4) 路径信息素更新策略

路径上的信息素为蚂蚁之间提供了间接的通信手段, 也就是说, 蚂蚁可以通过感知路径上的信息素浓度来完成彼此间的通信, 因此, 在蚁群算法完成一次循环后, 更新蚂蚁走过路径上的信息素浓度是至关重要的. 通常信息素更新可以采用如下方式:

$$\tau_{ij}^{new} = \rho \times \tau_{ij}^{old} + \sum_{k=1}^K \Delta \tau_{ij}^k, \quad \rho \in (0, 1) \quad (8)$$

式中,  $\tau_{ij}^{old}$ ,  $\tau_{ij}^{new}$  分别表示路段  $(i, j)$  上更新前后的路径信息素数量;  $\sum \Delta \tau_{ij}^k$  表示路段  $(i, j)$  上路径信息素的增量;  $k$  为解决方案中第  $k$  条路径序号;  $K$  表示解决方案中包含的路径数.

为了加快收敛速度以及使更好的方案在后续搜索中更加具有吸引力, 因此, 本文采用了一种 ant-weight 策略<sup>[12]</sup> 来计算路径信息素增量, 具体公式如下:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{K \times f^\theta} \times \frac{L^k - l_{ij}}{m^k \times L^k}, & \text{如果边}(i,j)\text{属于 } k_{th}\text{路径} \\ 0, & \text{其他} \end{cases} \quad (9)$$

式中,  $f$  表示解决方案的目标函数值 (所有路径的长度);  $l_{ij}$  表示第  $k$  条路径在路段  $(i, j)$  上的目标函数值 (长度).  $L^k$  表示解决方案中第  $k$  条路径的长度;  $m^k$  表示在解决方案中第  $k$  条路径中的客户数,  $m^k \geq 1$ ;  $\theta$  表示控制路径信息素增量的自适应因子,  $\theta \geq 1$ , 最初, 采用较大的路径信息素增量有利于提高进化初期的收敛速度, 而当进化的一定程度后, 减小路径信息素增量, 使算法可以进行详细的局部搜索, 如果  $\theta = 1$ , 则该策略即为静态的更新策略, 具体公式如下:

$$\theta = \lfloor t \times 2/T \rfloor + 1 \quad (10)$$

式中,  $\lfloor \cdot \rfloor$  表示向下取整符号, 例如,  $\lfloor 1.8 \rfloor = 1$ .

在 ant-weight 策略中, 每个方案总路径信息素增量为  $Q/f^\theta$ , 每条路径共分得  $Q/(K * f^\theta)$ , 每条路径所包含的每个路段的信息素增量则与其对于整个路径的贡献相关, 也就是说, 路段  $(i, j)$  上的路径信息素增量的比例为  $(L^k - l_{ij}) / (m^k \times L^k)$ .

### 3.3 程序流程

本文提出的两阶段启发式算法可以描述如下:

**BEGIN**

初始化运行环境及相关参数;

// 按配送中心分类客户点

计算客户相对于配送中心的分布矩阵, 初始化分类信息素矩阵;

**While** (不满足收敛条件)

**While** (仍有客户未选择配送中心)

从所有未分类客户随机选择一个客户;

计算该客户与每个配送中心当前客户集合的时间衔接性;

计算该客户与每个配送中心的聚集度;

计算该客户选择每个配送中心的概率;

通过轮盘赌方法选择配送中心;

**End while**

// 应用改进的蚁群算法求解单中心 VRPTW 问题

**While** (存在未优化的配送中心)

计算期望值矩阵, 初始化路径信息素矩阵;

**While** (不满足结束条件)

通过转移概率公式构造方案;

参考交叉率, 进行交叉操作及 2-opt 优化;

依据 ant-weight 策略更新路径信息素矩阵;

**End While**

**End while**

求和所有配送中心解决方案的目标函数值;

更新路径信息素矩阵;

**End while**

**End**

## 4 实例研究

为了验证提出的两阶段启发式算法的性能, 本文采用了若干经典的 MDVRPTW 实例对该算法进行了测试. 两阶段启发式算法的参数如表 1 所示. MDVRPTW 实例的具体信息如表 2 所示 (来源: <http://neumann.hec.ca/chairedistributique/data>).

表 1 参数设置

$\mu$	$\gamma$	$\rho$	$Q$	$\alpha$	$\beta$
1	3	0.95	2000	1	3

表 2 测试数据

实例	客户数	配送中心数	容量约束	已知最好解	实例	客户数	配送中心数	容量约束	已知最好解
1	48	4	200	1074.12	11	48	4	200	1005.73
2	96	4	195	1762.21	12	96	4	195	1478.51
3	144	4	150	2373.65	13	144	4	150	2011.24
4	192	4	185	2852.29	14	192	4	185	2202.08
5	240	4	180	3029.65	15	240	4	180	2494.57
6	288	4	175	3627.18	16	288	4	175	2901.02
7	72	6	200	1418.22	17	72	6	200	1236.24
8	144	6	190	2102.61	18	144	6	190	1792.61
9	216	6	180	2737.82	19	216	6	180	2285.10
10	288	6	170	3505.27	20	288	6	170	3079.16

### 4.1 客户分类算法及蚁群改进策略的性能比较

本文提出的两阶段启发式算法包含客户分类阶段和改进蚁群算法的线路优化阶段. 为了分析各算法和策略的性能, 本文构造了几种蚁群算法进行比较. 首先, 在启发式分类算法中, 如何计算确定性信息将直接关系到最终的分类结果. 因此, 我们引入一个基于客户相对于配送中心空间分布的启发式分类方法 (基于距离的启发式分类方法) 与本文提出的分类方法 (基于聚集度的启发式分类方法) 进行比较. 该分类方法在计算确定信息时, 仅考虑客户与配送中心的距离, 而不考虑客户间的时间衔接性, 其他与本文提出的分类方法相同. 这样, 我们构造如下几种蚁群算法: 带有基于距离的启发式分类方法的标准蚁群算法 (ACO-CD), 带有基于聚集度的启发式分类方法的标准蚁群算法 (ACO-CC), 带有基于聚集度的启发式分类方法的改进蚁群算法 (IACO-CC). 然后, 在相同的条件下, 对三种算法进行了 10 次测试, 计算结果如表 3 所示.

表 3 几种蚁群算法的比较

实例	IACO-CC	ACO-CC	ACO-CD	实例	IACO-CC	ACO-CC	ACO-CD
1	<b>1080.58</b>	1085.7	1086.85	11	<b>1006.23</b>	1007.99	1008.98
2	<b>1763.39</b>	1766.6	1768.96	12	<b>1480.45</b>	1486.3	1490.21
3	<b>2374.35</b>	2440.6	2655.38	13	<b>2011.93</b>	2222.3	2300.00
4	<b>2852.99</b>	3349.7	3605.49	14	<b>2221.45</b>	2568.99	2700.18
5	<b>3139.79</b>	3372.1	3785.49	15	<b>2502.52</b>	3079.87	3274.48
6	<b>3791</b>	4153.5	4604.92	16	<b>2951.81</b>	3387.01	3662.03
7	<b>1428.32</b>	1453.98	1474.76	17	<b>1251.37</b>	1287.5	1313.06
8	<b>2104.71</b>	2257.4	2454.02	18	<b>1813.67</b>	2029.23	2170.71
9	<b>2768</b>	3065.3	3448.29	19	<b>2305.7</b>	2665.63	2871.53
10	<b>3597.87</b>	3976.7	4441.08	20	<b>3098.1</b>	3678.89	3992.25

总体看, ACO-CD 算法的性能不如 ACO-CC 算法, 这体现了基于聚集度的分类更适合进行客户的分类操作. 特别是从表 3 可以发现, 对于前 10 个实例 ACO-CC 的优势更明显 (前 10 个实例 ACO-CC 优于 ACO-CD 大约 6.71%, 后 10 个实例为 4.55%). 这主要是因为, 前 10 个实例采用了比较严格的时间窗, 而后 10 个实例的时间窗比较宽松, 这进一步体现了采用聚集度对于带有时间窗的问题进行分类的必要性, 也体现了本文的分类算法的优越性. 另外, 我们也可以看到, IACO-CC 的性能要优于 ACO-CC, 特别是对于客户规

模较大的实例 (例如, 客户数量超过 150). 这主要是因为, ACO-CC 在进行线路优化时, 采用的是标准蚁群算法, 其容易出现早熟现象. 而在 IACO-CC 算法中, 交叉操作和自适应的信息素更新策略的引入有效的抑制了早熟的发生, 扩大了算法的搜索空间, 从而提高优化质量.

#### 4.2 两阶段启发式算法性能测试

为了进一步检验本文提出的两阶段启发式算法 (IACO-CC) 的性能, 本文将 IACO-CC 取得的最好解与 Polacek 等提出的近邻算法 (VNS)<sup>[1]</sup> 以及 Cordeau 等提出的禁忌算法 (TS)<sup>[2]</sup> 和改进的禁忌算法 (ITS)<sup>[3]</sup> 进行了比较, 结果如表 4 所示.

表 4 计算结果比较

实例	TS	VNS	ITS	IACO-CC	实例	TS	VNS	ITS	IACO-CC
1	1083.98	1083.98	<b>1074.12</b>	1080.01	11	1031.49	<b>1005.73</b>	<b>1005.73</b>	<b>1005.73</b>
2	1763.07	<b>1762.21</b>	<b>1762.21</b>	<b>1762.21</b>	12	1500.48	1487.64	<b>1478.51</b>	<b>1478.51</b>
3	2408.42	2374.36	<b>2373.65</b>	<b>2373.65</b>	13	2020.58	2014.02	<b>2011.24</b>	<b>2011.24</b>
4	2958.23	2858.2	<b>2852.29</b>	<b>2852.29</b>	14	2247.72	2221.17	<b>2202.08</b>	2217.09
5	3134.04	3040.84	<b>3029.65</b>	3134.04	15	2509.75	2494.82	<b>2494.57</b>	<b>2494.57</b>
6	3904.07	3758.36	<b>3627.18</b>	3758.36	16	2943.9	2939.2	<b>2901.02</b>	2941.56
7	1423.35	1422.65	<b>1418.22</b>	<b>1418.22</b>	17	1250.09	1239.13	<b>1236.24</b>	1239.13
8	2150.22	2103.89	<b>2102.61</b>	2103.89	18	1809.35	1796.21	<b>1792.61</b>	<b>1796.21</b>
9	2833.8	2783.95	<b>2737.82</b>	2753.21	19	2310.92	2318.59	<b>2285.10</b>	2291.98
10	3717.22	3577.28	<b>3505.27</b>	3582.82	20	3131.9	3079.73	<b>3079.16</b>	<b>3079.16</b>

从表 4 中的结果可知, 本文提出的两阶段启发式算法的优化质量明显高于 TS 和 VNS 两种算法. 本文提出的算法在所有的 20 个实例中有 10 个实例达到了已知的最好解. 而且, 从优化结果看, 本文算法在求解相对规模较小的实例时有更好的表现. 总体看, IACO-CC 算法的优化质量不如 ITS 算法, 但是, 两种算法的优化结果相差不足 1%, 表明本文提出的两阶段启发式算法仍然是一种有竞争力的算法.

## 5 结论

车辆路径问题是物流研究领域中的一个具有十分重要理论和现实意义的问题. 由于配送路径可以是所有客户的任意组合, 该问题通常被认为是 NP-hard 问题. 而带时间窗的多中心的路径优化问题可以看成是单中心路径优化问题的一个扩展, 因此其通常被认为更加难于求解. 本文首先提出构建基于聚集度的启发式分类算法将 MDVRPTW 简化为多个单中心 VRPTW, 然后, 即可以使用成功应用于单中心车辆路径问题的算法对该问题进行求解. 蚁群算法是一种相对较新的进化算法, 本文提出两种改进策略来提高蚁群算法的性能: 交叉策略和自适应 ant-weight 策略. 最后, 通过若干经典的 MDVRPTW 对该算法进行了验证, 结果显示两阶段启发式算法是一种比较有效的求解算法.

## 参考文献

- [1] Polacek M R, Hartl R F, Doerner K, et al. A variable neighborhood search for the multi depot vehicle routing problem with time windows[J]. *Journal of Heuristics*, 2004, 10(6): 613-627.
- [2] Cordeau J F, Laporte G, Mercier A. A unified tabu search heuristic for vehicle routing problems with time windows[J]. *Journal of the Operational Research Society*, 2001, 52(8): 928-936.
- [3] Cordeau J F, Laporte G, Mercier A. An improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows[J]. *Journal of the Operational Research Society*, 2004, 55(5): 542-546.
- [4] Renaud J, Laporte G, Boctor F F. A tabu search heuristic for the multi-depot vehicle routing problem[J]. *Computers and Operations Research*, 1996, 23(3): 229-235.
- [5] Cordeau J F, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems[J]. *Networks*, 1997, 30: 105-119.
- [6] Chao M I, Golden B L, Wasil E A. A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions[J]. *Am J Math Mgmt Sci*, 1993, 13: 371-406.
- [7] Christofides N, Eilon S. An algorithm for one vehicle-dispatching problem[J]. *Opl Res Q*, 1969, 20: 309-318.

- [8] Gillett B E, Johnson J G. Multi-terminal vehicle-dispatch algorithm[J]. *Omega*, 1976(4): 711–718.
- [9] Crevier B, Cordeau J F, Laporte G. The multi-depot vehicle routing problem with inter-depot routes[J]. *European Journal of Operational Research*, 2007, 76(2): 756–773.
- [10] 王平, 唐喜平, 李云. 一类多源点物流配送优化模型的探讨 [J]. *系统工程理论与实践*, 2003, 23(3): 87–91.  
Wang P, Tang X P, Li Y. Approach for a multi-source logistics delivery optimum model[J]. *Systems Engineering — Theory & Practice*, 2003, 23(3): 87–91.
- [11] 李臻, 雷定猷. 多车场车辆优化调度模型及算法 [J]. *交通运输工程学报*, 2004, 4(1): 83–86.  
Li Z, Lei D Y. Model and algorithm of multiple depot transit vehicle scheduling[J]. *Journal of Traffic and Transportation Engineering*, 2004, 4(1): 83–86.
- [12] 李敏, 郭强, 刘红丽. 多车场多配送中心的物流配送问题研究 [J]. *计算机工程与应用*, 2007, 43(8): 202–204, 208.  
Li M, Guo Q, Liu H L. Multiple depot multi-logistics center distribution problem[J]. *Computer Engineering and Applications*, 2007, 43(8): 202–204, 208.
- [13] 李相勇, 田澎. 带时间窗和随机时间车辆路径问题: 模型和算法 [J]. *系统工程理论与实践*, 2009, 29(8): 81–90.  
Li X Y, Tian P. Vehicle routing problems with time windows and stochastic times: Models & algorithm[J]. *Systems Engineering — Theory & Practice*, 2009, 29(8): 81–90.
- [14] 丁秋雷, 胡祥培, 李永先. 求解有时间窗的车辆路径问题的混合蚁群算法 [J]. *系统工程理论与实践*, 2007, 27(10): 98–104.  
Ding Q L, Hu X P, Li Y X. A hybrid ant colony system for vehicle routing problem with time windows[J]. *Systems Engineering — Theory & Practice*, 2007, 27(10): 98–104.
- [15] 钟石泉, 马寿峰. 车辆路径问题的改进分支切割法 [J]. *系统工程理论与实践*, 2009, 29(10): 152–158.  
Zhong S Q, Ma S F. Improved branch and cut algorithm for vehicle routing problem[J]. *Systems Engineering — Theory & Practice*, 2009, 29(10): 152–158.
- [16] Dorigo M, Maniezzo V, Colomi A. The ant system: Optimization by a colony of cooperating agents[J]. *IEEE Transactions on Systems, Mans, and Cybernetics*, 1996, 26(1): 29–41.
- [17] Yu B, Yang Z Z, Yao B Z. An improved ant colony optimization for vehicle routing problem[J]. *European Journal of Operational*, 2009, 196(1): 171–176.
- [18] Bullnheimer B, Hartl R F, Strauss C. An improved ant system algorithm for the vehicle routing problem[J]. *Annals of Operations Research*, 1999, 89: 319–328.