

# 基于启发式带时间窗的车辆路径规划问题求解<sup>①</sup>

刘砾佳, 李学强

(东莞理工学院 计算机科学与技术学院, 东莞 523808)

通信作者: 李学强, E-mail: lixq@dgut.edu.cn



**摘要:** 车辆路径规划问题广泛地存在于现代物流行业中, 该问题属于 NP 难的组合优化问题。随着客户需求的多样化、道路限行等因素的影响, 该问题变得更加复杂, 采用传统的组合优化方法和运筹学方法往往难以求解。本文对一类常见的带时间窗的车辆路径规划问题进行了研究, 根据时间窗参数来调整客户的优先级, 以减少车辆的等待时间, 由此改进了几个常见的启发式算法, 并对 56 个常见的车辆路径规划问题进行了测试, 实验结果表明, 改进的节约算法在带容量约束的车辆路径问题中效果较好, 改进的插入法则在带时间窗的车辆路径问题中具有优越性, 另外, 改进的启发式算法在 4 个测试用例上使用更多车辆时可使总路程优于已知最优值。

**关键词:** 启发式算法; 带时间窗的车辆路径问题; 节约算法; 插入法

引用格式: 刘砾佳, 李学强. 基于启发式带时间窗的车辆路径规划问题求解. 计算机系统应用, 2022, 31(11):275–281. <http://www.c-s-a.org.cn/1003-3254/8818.html>

## Solution to Vehicle Routing Problem with Time Windows Based on Heuristic Algorithm

LIU Shuo-Jia, LI Xue-Qiang

(School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China)

**Abstract:** The vehicle routing problem (VRP) exists extensively in the modern logistics industry, which is an NP-hard problem in combinatorial optimization. Affected by factors such as diverse customer demand and road traffic restrictions, VRP becomes more complex, and it can hardly be solved by the traditional combinatorial optimization methods and operations research methods. In this study, a common VRP with time windows is studied. The waiting time of vehicles is reduced by the adjustment to the priority of customers according to the parameters of time windows. On this basis, several common heuristic algorithms are improved, and 56 common VRPs are tested. The experimental results reveal that the improved savings algorithm can produce good results for capacitated VRPs, and the improved insertion method has superior performance in VRPs with time windows. In addition, the improved heuristic algorithms can make the total distance better than the known optimal value when using more vehicles on the four test cases.

**Key words:** heuristic algorithm; vehicle routing problem with time windows; saving algorithm; insertion method

车辆路径规划问题 (vehicle routing problem, VRP) 最早于 1959 年提出<sup>[1]</sup>, 该问题可以描述为: 对于一组客户点以及一定数量的车辆, 在满足一定的约束条件的情况下, 对处于不同物理位置上的大规模客户进行服务, 并使得总成本最小。大多数的物流运输问题都可以看成是 VRP 问题, 该问题根据约束条件的不同, 可分为 CVRP、VRPTW、DCVRP、VRPB 等不同类型的

问题。其中, CVRP (capacitated vehicle routing problem) 是一种经典的车辆路径问题, 每辆车都有相同的容量限制, 车辆都从同一个配送中心出发并最终返回到配送中心, 且已知各个客户点的位置和需求量; 带时间窗的车辆路径问题 (VRP with time windows, VRPTW) 则是在 CVRP 问题的基础上增加了时间窗约束, 车辆必须在规定时间窗内到达客户点并提供服务。

① 基金项目: 广东省“创新强校”工程项目 (GB200101-41); 东莞理工学院校级科研团队物联网系统及应用团队项目 (TDYB2019009)

收稿时间: 2022-03-08; 修改时间: 2022-04-07; 采用时间: 2022-04-22; csa 在线出版时间: 2022-07-14

车辆路径问题通常使用精确算法或启发式算法进行求解。其中精确算法包括分支定价法<sup>[2-4]</sup>、割平面法<sup>[5-6]</sup>和动态规划法<sup>[7]</sup>等。然而,该类算法通常应用于小规模的车辆路径规划问题,当客户点的数量较大时,该类精确算法就会失效。文献[8]提出了一种标签法,解决了超过15个此前未得到最优解的Solomon测试例,但部分测试例的求解时间超过了10个小时。因此,精确算法很难用于求解大规模的车辆路径规划问题。车辆路径问题属于NP难的组合优化问题,目前对于大规模的车辆路径规划问题一般采用启发式算法求解。其中元启发式算法是求解VRP问题常用的启发式算法,这类算法包括遗传算法、蚁群算法、模拟退火算法、禁忌搜索等<sup>[9-14]</sup>。改进型启发式算法,如2-opt、3-opt等<sup>[15]</sup>,通过在路径内部或路径之间的交换操作改进解的质量。构造型启发式法则通过逐步构造路径来形成可行解,常见的构造型启发式算法有节约算法<sup>[16]</sup>、最近邻法<sup>[17]</sup>、扫描法<sup>[18]</sup>等。

对于复杂的VRP,采用单一的启发式算法往往很难得到令人满意的结果,目前,通常采用混合式的启发式算法来求解该问题。该类算法首先通过采用合理的启发式算法来构造车辆路径问题的初始解,然后采用群体智能搜索模式,如遗传算法、蚁群算法、模拟退火算法、禁忌搜索等,在初始解的基础上对复杂的车辆路径规划问题进行求解。由于该类组合优化问题属于多变量多陷阱的最优解的问题,因此,初始解的生成质量,往往也影响着群体搜索算法的搜索效率和最终解的质量。基于此,本文对几种启发式算法进行了实验分析和改进,并将其用于求解现实生活中常见的一类带时间窗的车辆路径问题,以期望得到更好的初始解,最后,在常用的一些关于车辆路径规划问题的数据集上对改进的启发式算法进行了比较,并对结果进行了分析。

## 1 问题描述及数学模型

VRPTW问题可以表述为:配送中心给定 $m$ 辆容量为 $Q$ 的车辆,为 $n$ 个客户提供服务。其中,车辆集 $V=\{1, 2, 3, \dots, m\}$ ,客户集 $C=\{0, 1, 2, \dots, n\}$ ,客户0表示配送中心,客户 $i$ 的货物需求量表示为 $q_i$ ( $q_0=0$ );从客户 $i$ 到客户 $j$ 的距离为 $d_{ij}$ ,所耗费的行驶时间为 $t_{ij}$ ,在客户 $i$ 上花费的服务时间为 $s_i$ ;客户 $i$ 所允许的服务时间窗为 $[a_i, b_i]$ ,则车辆 $k$ 到达客户 $i$ 的时间 $r_{ik}$ 应满足

$r_{ik} \in [a_i, b_i]$ 。车辆从配送中心出发,为每个客户提供服务,最后返回到配送中心,要求规划运输路径,使得运输成本最小。因此,VRPTW数学模型可表示为:

$$\min Z = \sum_{i \in C} \sum_{j \in C} \sum_{k \in V} d_{ij} x_{ijk} \quad (1)$$

$$\sum_{j \in C} \sum_{k \in V} x_{ijk} = 1, \forall i \in C - \{0\} \quad (2)$$

$$\sum_{i \in C} x_{ihk} - \sum_{j \in C} x_{hjk} = 0, \forall h \in C, \forall k \in V \quad (3)$$

$$\sum_{j \in C} x_{0jk} = 1, \forall k \in V \quad (4)$$

$$\sum_{i \in C} q_i \sum_{j \in C} x_{ijk} \leq Q, \forall k \in V \quad (5)$$

$$r_{ik} + t_{ij} + s_i - K(1 - x_{ijk}) \leq r_{jk}, \forall i, j \in C, \forall k \in V \quad (6)$$

$$a_i \leq r_{ik} \leq b_i, \forall i \in C, \forall k \in V \quad (7)$$

$$x_{ijk} \in (0, 1), \forall i, j \in C, \forall k \in V \quad (8)$$

式(1)表示最小化车辆总行驶路程;式(2)和式(3)表示每个客户均有一辆车提供服务且只能被服务一次;式(4)约束了车辆从配送中心出发服务客户并且只出发一次;式(5)是对车辆容量的约束;式(6)和式(7)是车辆到达各个客户的时间窗约束,其中 $K$ 是一个足够大的常数。

## 2 传统的启发式算法

CW(Clarke Wright)节约算法是解决车辆路径问题最常用的启发式算法,其核心思想是通过合并路径减少总路程<sup>[16]</sup>。在开始时,由 $n$ 辆车为 $n$ 个客户点提供服务,形成 $n$ 条初始路径。如果两条路径 $(0, \dots, i, 0)$ 和 $(0, j, \dots, 0)$ 合并成一条路径 $(0, \dots, i, j, \dots, 0)$ 后,不违反车辆的容量约束,那么合并后的路径将产生节约值: $s_{ij} = d_{i0} + d_{0j} - d_{ij}$ 。节约值 $s_{ij}$ 表示路径合并后节约的路程。CW算法优先对节约值大的路径进行合并,直到不存在可进行合并的路径时停止。

最近邻法通过每次选择距上一个客户点最近的客户点来构造路径<sup>[17]</sup>。首先,选择配送中心作为路径起点,之后每次选择与上一个客户点最近邻的客户点加入路径。当将客户点加入路径后不满足车辆容量约束时,开始一条新的路径,重复过程至所有客户点均被服务。

扫描法是一种“先分组后排线”的启发式算法<sup>[18]</sup>. 分组时,首先从配送中心引出一条射线,按顺时针或逆时针顺序扫描,将扫描到的客户点依次加入路径.当客户点加入路径后客户总需求量超出车辆容量时,开始规划一条新路径,重复该过程直至所有客户点均被服务.排线时,使用精确算法或2-opt等启发式算法对各组客户点重新排线或进一步优化.

插入法在构造路径时,根据特定的原则选取初始路径的第一个客户点,而后选择最佳的客户点插入最佳插入位置<sup>[19]</sup>.为了减少计算次数,可以使用启发式算法构造一个包含所有客户点的初始序列,并初始化一条空路径,依次从初始序列中取出客户点插入到初始路径的最佳插入位置,使得总路程增加最少.当插入客户点后总需求量超过车辆容量时,开始规划新路径.最后,考虑到在构造路径时,插入法仅在初始路径中寻找客户点的最佳插入位置,因此对每个客户点,尝试将之插入到后续构建的路径中以减少总路程.

### 3 考虑时间窗的启发式算法

在上述4种传统启发式算法中引入时间窗约束.传统的启发式算法在路径末尾添加客户点时由于不需要考虑时间窗,因此很容易出现车辆到达了下一个客户点而时间窗尚未开始、车辆被迫等待的情况,这必然会减少当前车辆服务的客户点数量.为此,本文的算法会优先将可能使时间窗更紧凑的客户点加入路径,以减少等待时间.如果客户点*i*、*j*满足 $b_i + s_i + d_{ij} > a_j$ ,则有可能访问客户点*i*后访问客户点*j*时,没有或只要少许的等待时间.其中, $b_i$ 、 $s_i$ 分别是客户点*i*时间窗的结束时间和服务时间, $d_{ij}$ 是从客户点*i*到客户点*j*经过的路程, $a_j$ 是客户点*j*时间窗开始的时间.本文以此作为判断考虑是否优先将客户加入路径.

#### 1) CW 算法

改进的CW算法优先考虑了可能使时间窗更紧凑的子路径,重新定义客户点*i*、*j*的节约值为:如果 $b_i + s_i + d_{ij} > a_j$ ,则 $s_{ij} = d_{i0} + d_{0j} - d_{ij} + N$ ,其中N是一个大的正数;对于其他情况,仍是 $s_{ij} = d_{i0} + d_{0j} - d_{ij}$ .算法步骤如下.

Step 1. 计算两两客户点之间的节约量 $s_{ij}$ , $i, j \in C$ , $C = \{1, 2, 3, \dots, n\}$ ,并将节约量按从大到小的顺序排序.

Step 2. 初始化n条路径,每条路径中只有一个客户点,起点和终点为配送中心.

Step 3. 依次选取节约量 $s_{ij}$ ,如果客户点*i*、*j*分别

位于路径的开始和末尾处,且二者不在同一条路径上,进入Step 4;否则,依Step 3重新选取节约量.当所有节约量均被选取时,算法终止.

Step 4. 如果路径 $(0, \dots, i, 0)$ 和 $(0, j, \dots, 0)$ 合并为路径 $(0, \dots, i, j, \dots, 0)$ 后不违反车辆的容量约束和时间窗约束,则将该路径合并,然后返回Step 3;否则,直接返回Step 3,重新选取节约量.

#### 2) 最近邻法

与CW算法的改进相似,改进的最近邻法优先将时间窗更紧凑距离更近的客户点加入路径.具体的,如果不满足条件 $b_i + s_i + d_{ij} > a_j$ ,则将客户点*i*、*j*之间的距离定义为:真实距离加上一个大的正数N.改进的算法步骤如下.

Step 1. 初始化一条空路径,将配送中心作为起点加入路径.

Step 2. 计算客户集C中的客户点到路径最后一个客户点的距离,如果在客户集C中存在加入路径末尾后不违反车辆容量约束和时间窗约束的客户点,则进入下一步;否则,转至Step 4.

Step 3. 在不违反约束的客户点中选取距离最小的加入到路径末尾,并在客户集C中移除该客户点.返回Step 2.

Step 4. 将配送中心加入到路径末尾.如果客户集C不为空,则算法返回至Step 1,开始安排新的路径;否则,所有客户均已被服务,算法终止.

#### 3) 扫描法

在随机选定起始客户点*i*后,优先将满足 $b_i + s_i + d_{ij} > a_j$ 的客户点插入路径.采用此策略时,单向的扫描将遗漏部分客户点,因而本文采用的是双向扫描的策略,即从当前加入路径的客户点开始,同时进行顺、逆时针的扫描.同时,由于车辆到达起始客户点时可能存在较长的等待时间,因而将扫描到的客户点加入到起始或末尾位置.改进后的算法步骤如下.

Step 1. 初始化一条空路径,将配送中心作为起点加入路径.从客户集C中任选一个客户点加入路径.

Step 2. 以路径中最后一个客户点*i*与配送中心的连线为始边,计算配送中心与各客户点连线的角度.如果客户*j*不满足 $b_i + s_i + d_{ij} > a_j$ ,则将角度值加上一个大的正数N.

Step 3. 当客户集C不为空时,选择客户集C中角度最小的客户点,若加入路径起始或末尾位置后不违

反车辆容量约束和时间窗约束, 转至 Step 4; 否则, 转至 Step 5.

Step 4. 将该客户点加入路径的起始或末尾位置, 并从客户集  $C$  中移除该客户点, 返回 Step 2.

Step 5. 将配送中心加入路径末尾, 表示车辆返回配送中心. 若客户集  $C$  不为空, 则开始规划一条新路径, 返回 Step 1; 否则, 所有客户均已被服务, 算法终止.

#### 4) 插入法

插入法会尝试将客户点插入到路径中间, 因而不需要考虑时间窗是否紧凑的问题. 在此, 仅考虑将客户点插入到满足时间窗约束的位置. 假定初始序列为  $(c_1, c_2, \dots, c_n)$ , 算法步骤如下.

Step 1. 令  $j=1, k=2$ , 并初始化路径  $route_j=(0, c_1, 0)$ .

Step 2. 如果客户点  $c_k$  插入路径  $route_j$  后总需求量超过车辆容量, 则开始规划一条新的车辆路径  $route_{j+1}=(0, c_k, 0)$ , 并令  $j=j+1$ , 转至 Step 4. 否则, 进入下一步.

Step 3. 对路径  $route_j$  中所有可以插入客户点  $c_k$  而不违反时间窗约束的位置, 计算插入代价. 将  $c_k$  插入路径  $(0, \dots, i, j, \dots, 0)$  中  $i, j$  之间的位置时, 其插入代价为  $cost_{ij}=d_{ick}+d_{ckj}-d_{ij}$ . 若不存在合适的插入位置, 则规划一条新的车辆路径  $route_{j+1}=(0, c_k, 0)$ , 并令  $j=j+1$ , 转至 Step 4. 否则, 选择插入代价最小的位置插入客户点  $c_k$ , 并转至 Step 4.

Step 4.  $k=k+1$ . 若  $k \leq n$ , 返回 Step 2. 否则进入下一步.

Step 5. 设所规划的路径数量为  $J$ , 路径  $route_j$  经过的客户点数量为  $n_j$ , 且  $route_j=(0, r_1, r_2, \dots, r_{n_j}, 0), j=1, 2, \dots, J$ ; 令  $j=J$ .

Step 6. 令  $i=n_j$ .

Step 7. 对客户点  $r_i$ , 计算路径  $(route_{j+1}, route_{j+2}, \dots, route_J)$ , 以及  $route_j$  的部分路径  $(r_{i+1}, r_{i+2}, \dots, r_{n_j}, 0)$  中的插入代价. 在满足车辆容量约束和时间窗约束的插入位置中, 选择插入代价最小的位置插入客户点  $r_i$ .

Step 8. 令  $i=i-1$ . 若  $i > 0$ , 转至 Step 7.

Step 9. 令  $j=j-1$ . 若  $j > 0$ , 转至 Step 6; 否则算法终止.

## 4 仿真实验

### 4.1 实验数据

实验采用 CMT 数据集以及 Solomon 数据集进行测试. CMT 数据集由 14 个测试例构成, 客户点数量在

50 到 200 之间. 其中 CMT1, CMT2, CMT3, CMT4, CMT5, CMT11 以及 CMT12 是属于 CVRP 的测试例, CMT6, CMT7, CMT8, CMT9, CMT10, CMT13 以及 CMT14 则在 CVRP 的基础上对每个客户点添加了服务时间, 对每辆车限制了最大行驶时长. Solomon 数据集带有时间窗, 是 VRPTW 的一个标准测试集, 由 56 个测试例构成, 每个测试例有 100 个客户点, 测试例可分为 6 大类, 分别是 R1 类、R2 类、C1 类、C2 类、RC1 类和 RC2 类. 其中, R 类的客户点坐标分布均匀, C 类问题的客户点呈聚集性分布, RC 类的测试例同时包含了均匀分布的客户点和呈聚集性分布的客户点. 此外, C1 类、R1 类和 RC1 类具有较窄的时间窗和较小的车辆容量, 而剩余的 3 类问题则具有较宽的时间窗和更大的车辆容量.

### 4.2 实验设置

实验使用 Matlab 进行编程实现, 分别将 CW 算法、最近邻法、扫描法以及插入法对 CMT 以及 Solomon 数据集的每个测试用例进行了实验. 其中 CMT 数据集的部分测试用例属于带行驶时间限制的车辆路径问题 (duration constrained CVRP), 对于这部分测试例, 将上述各算法对时间窗的约束改为对行驶时长的约束.

对该问题求解过程中本文设置了以下的一些假设:

(1) 各客户点之间的距离采用欧氏距离, 并以双精度的浮点数表示.

(2) 假定车辆的行驶速率恒定, 并且车辆的行驶时间大小等于行驶距离.

(3) 插入法的初始序列由扫描法给出, 扫描法的初始客户点随机给出.

(4) 为了方便比较, 扫描法不使用 3-opt 等改进算法.

(5) 确定性算法 CW 算法以及最近邻法对每个测试例仅测试一次, 随机算法包括扫描法和插入法则对每个测试例都进行 150 次测试.

### 4.3 实验结果与分析

4 种算法在 CMT 数据集上的测试结果如表 1 所示. 对于 CW 算法、最近邻法, 表 1 给出了运行结果的精确值, 对于扫描法和插入法, 则给出在 150 次测试中得到的平均结果. 如表 1 所列, CW 算法和插入法的效果明显优于其余两种算法. 在所有测试例上最近邻法明显优于扫描法. 为了进一步验证 CW 算法在 CMT 数据集上是否显著优于其他算法, 本文使用配对样本 t 检

验将 CW 算法同其他算法进行比较，并在表 1 底部给出显著性分析结果 (Y 表示 CW 算法显著优于该算

法) 和对应的  $p$  值。可以看到，CW 算法在 CMT 数据集上显著优于其他算法。

表 1 4 种算法在 CMT 数据集上的测试结果

测试例	节约算法		最近邻法		扫描法		插入法	
	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)
CMT1	6.00	584.64	5.00	711.50	5.00	960.24	5.77	586.57
CMT2	11.00	914.27	10.00	1205.85	10.05	1591.49	11.00	926.33
CMT3	8.00	877.93	8.00	1174.36	8.00	1675.17	8.00	907.87
CMT4	12.00	1111.62	12.00	1426.23	12.00	2445.18	12.00	1150.28
CMT5	17.00	1422.01	17.00	2034.40	16.62	3079.05	17.04	1460.68
CMT6	6.00	622.63	7.00	803.03	8.27	1015.49	6.25	598.62
CMT7	12.00	972.39	14.00	1227.79	15.85	1605.09	12.05	995.71
CMT8	9.00	963.67	11.00	1232.75	12.61	1720.04	9.66	963.52
CMT9	15.00	1300.82	16.00	1562.72	21.21	2535.81	15.17	1288.42
CMT10	19.00	1575.80	20.00	1713.61	26.78	3127.45	19.31	1579.58
CMT11	7.00	1071.95	7.00	1565.34	7.00	3611.69	7.75	1402.15
CMT12	10.00	838.87	10.00	1311.50	10.00	1446.32	10.00	1013.87
CMT13	11.00	1588.40	12.00	1904.05	14.39	3751.22	12.00	1981.47
CMT14	11.00	877.06	11.00	1127.01	11.01	1475.93	11.00	1045.40
$T$ 检验 ( $\alpha=0.05$ )	—	—	6.76E-07	—	1.99E-05	—	1.68E-02	—

表 2 给出了 4 种算法在 Solomon 数据集上的部分测试结果以及已知的最优结果，其中扫描法和插入法给出的是在 150 次测试中得到的最优结果，已知最优解来自文献 [20]。如表 2 所列，节约算法和插入法更接近已知最优解，但 CW 算法所需要的车辆数较多，而插入法除了 R1 以及 RC1 之外，所需的车辆数量更接近已知最优解。同时，应注意 CW 算法在测试例 C101

上达到了已知最优解，插入法在测试例 C101, C105, C107 以及 C201 上得到了与已知最优解一样的结果。在测试例 R202, R211, RC201 以及 RC202 上，插入法所需的车辆数多于已知最优解，但在路程上得到了优于已知最优解的结果，这表明插入法更可能得到接近最优解的结果。图 1-图 6 给出了插入法在部分测试例上得到的路径。

表 2 4 种算法在 Solomon 数据集部分测试例上的运行结果

测试例	节约算法		最近邻法		扫描法		插入法		已知最优结果	
	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)
C101	10.00	828.94	10.00	878.36	14.00	1128.98	10.00	828.94	10.00	828.94
C105	11.00	934.28	14.00	1442.10	15.00	1333.30	10.00	828.94	10.00	828.94
C107	11.00	934.28	14.00	1558.05	13.00	1486.89	10.00	828.94	10.00	828.94
C201	5.00	774.63	8.00	1024.72	7.00	1275.70	3.00	591.56	3.00	591.56
C206	4.00	671.44	5.00	860.69	5.00	1137.37	4.00	635.69	3.00	588.49
C208	4.00	728.60	4.00	990.04	4.00	1024.77	4.00	627.49	3.00	588.32
R104	15.00	1152.70	17.00	1466.98	14.00	1621.21	11.00	1097.11	9.00	1007.31
R109	16.00	1346.31	19.00	1818.12	16.00	1745.38	16.00	1389.61	11.00	1194.73
R110	15.00	1271.86	17.00	1753.69	14.00	1666.40	15.00	1315.13	10.00	1118.84
R201	7.00	1445.73	11.00	1956.46	7.00	1946.72	6.00	1255.12	4.00	1252.37
R202	9.00	1310.54	13.00	1914.03	7.00	1699.08	5.00	1177.23	3.00	1191.70
R211	9.00	903.17	5.00	1268.64	3.00	1267.04	4.00	878.16	2.00	885.71
RC101	19.00	1976.45	25.00	2630.69	20.00	2196.87	18.00	1839.79	14.00	1696.95
RC103	17.00	1607.55	23.00	2186.07	16.00	1844.09	14.00	1510.84	11.00	1261.67
RC108	13.00	1240.26	15.00	1792.98	13.00	1722.48	14.00	1412.97	10.00	1139.82
RC201	7.00	1601.21	10.00	2341.10	7.00	2153.03	6.00	1401.32	4.00	1406.94
RC202	11.00	1372.00	10.00	2331.11	7.00	2115.93	5.00	1336.46	3.00	1365.65
RC208	7.00	883.07	5.00	1395.02	4.00	1547.52	4.00	1002.68	3.00	828.14

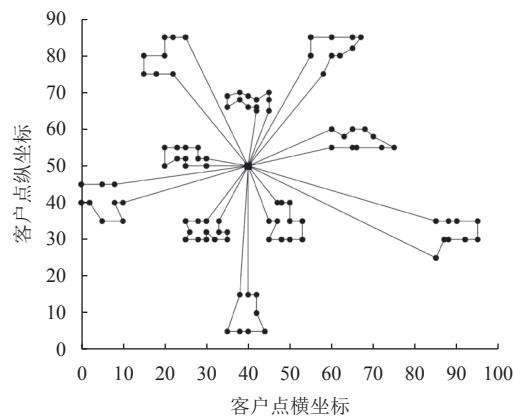


图1 插入法得到的C107测试例路径

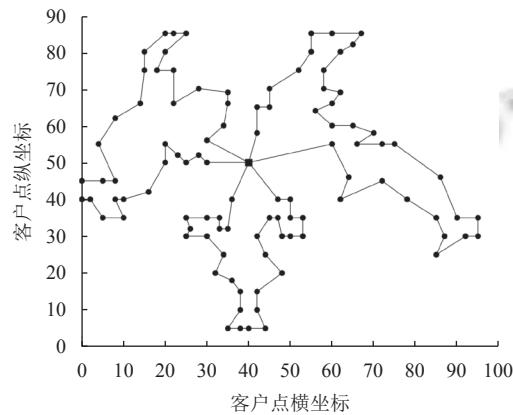


图2 插入法得到的C201测试例路径

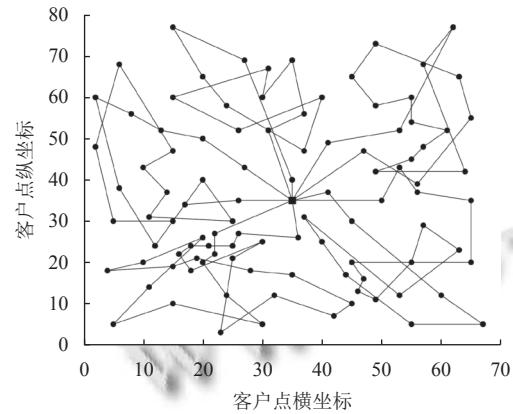


图3 插入法得到的R202测试例路径

表3列出了4种算法在6种类型的测试用例上的平均结果。结果显示，节约算法和插入法明显优于其他算法，扫描法表现最差。但不同于在CMT数据集上的测试结果的是，插入法在含有均匀分布客户点的R类及RC类测试例上得到了比节约算法更好的结果。但对于客户点呈聚集性分布的C类测试用例，改进的节约算法则明显优于改进的插入法。

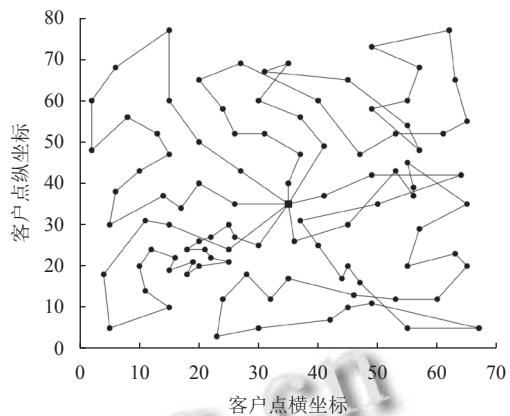


图4 插入法得到的R211测试例路径

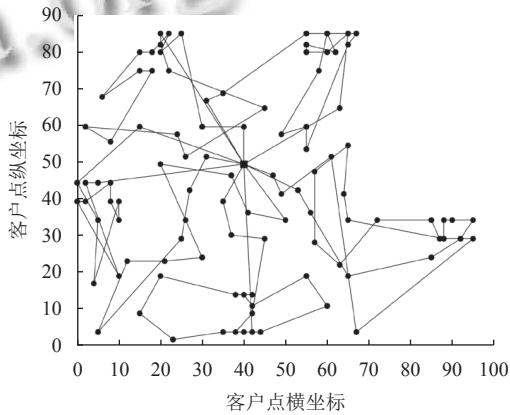


图5 插入法得到的RC201测试例路径

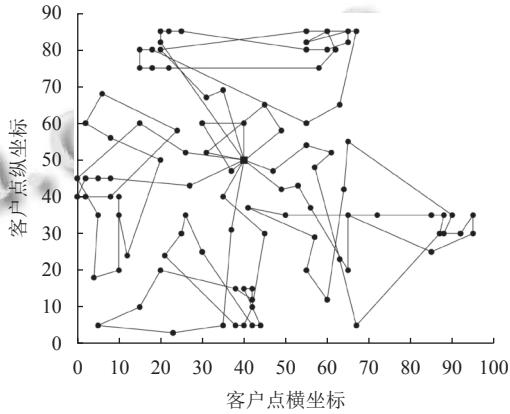


图6 插入法得到的RC202测试例路径

## 5 结论

本文针对带时间窗的车辆路径规划问题，基于时间窗来调整客户插入优先级，以此改进了传统的启发式算法，并分别在CVRP以及VRPTW的相关数据集上测试。结果表明，改进的CW算法和插入法均能获得较好的结果。并且，在CVRP的相关数据集上，CW算

法优于其他算法;而在VRPTW的相关数据集上,改进的插入法对含有均匀分布客户点的测试例效果最好,

改进的CW法则在客户点均匀分布的测试例上有更好的表现。

表3 4种算法在Solomon数据集上的平均测试结果

类型	节约算法		最近邻法		扫描法		插入法	
	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)	车辆(辆)	路程(km)
C1	10.89	931.21	14.11	1575.19	14.21	1776.96	11.21	1085.36
C2	4.88	742.79	7.25	1121.89	6.47	1525.00	4.46	762.32
R1	15.67	1386.43	22.58	1910.21	18.18	1987.40	16.53	1197.96
R2	7.55	1061.67	7.73	1477.51	5.43	1711.27	4.02	963.13
RC1	16.63	1601.82	21.00	2233.71	17.71	2247.85	16.66	1486.62
RC2	8.25	1218.71	8.75	1908.01	6.32	2143.54	4.74	1135.24

### 参考文献

- Dantzig GB, Ramser JH. The truck dispatching problem. *Management Science*, 1959, 6(1): 80–91. [doi: [10.1287/mnsc.6.1.80](https://doi.org/10.1287/mnsc.6.1.80)]
- Taş D, Gendreau M, Dellaert N, et al. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, 2014, 236(3): 789–799. [doi: [10.1016/j.ejor.2013.05.024](https://doi.org/10.1016/j.ejor.2013.05.024)]
- Dabia S, Ropke S, van Woensel T, et al. Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science*, 2013, 47(3): 380–396. [doi: [10.1287/trsc.1120.0445](https://doi.org/10.1287/trsc.1120.0445)]
- Hernandez F, Feillet D, Giroudeau R, et al. Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 2016, 249(2): 551–559. [doi: [10.1016/j.ejor.2015.08.040](https://doi.org/10.1016/j.ejor.2015.08.040)]
- Mak V, Ernst AT. New cutting-planes for the time-and/or precedence-constrained ATSP and directed VRP. *Mathematical Methods of Operations Research*, 2007, 66(1): 69–98. [doi: [10.1007/s00186-006-0141-x](https://doi.org/10.1007/s00186-006-0141-x)]
- Kallehauge B, Larsen J, Madsen OBG. Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 2006, 33(5): 1464–1487.
- Mahmoudi M, Zhou XS. 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. *Transportation Research Part B: Methodological*, 2016, 89: 19–42. [doi: [10.1016/j.trb.2016.03.009](https://doi.org/10.1016/j.trb.2016.03.009)]
- Irnich S, Villeneuve D. The shortest-path problem with resource constraints and  $k$ -cycle elimination for  $k \geq 3$ . *Informs Journal on Computing*, 2006, 18(3): 391–406. [doi: [10.1287/ijoc.1040.0117](https://doi.org/10.1287/ijoc.1040.0117)]
- 范厚明, 刘鹏程, 吴嘉鑫, 等. 集货需求随机的同时配集货 VRP 及混合变邻域搜索算法. *系统工程理论与实践*, 2019, 39(10): 2646–2659. [doi: [10.12011/1000-6788-2018-0957-14](https://doi.org/10.12011/1000-6788-2018-0957-14)]
- 王丹. 基于改进遗传算法的物流配送车辆路径问题研究 [硕士学位论文]. 长春: 长春工业大学, 2016.
- 王雪红. 基于遗传算法的车辆路径优化问题的应用研究 [硕士学位论文]. 天津: 天津科技大学, 2016.
- 孙小军, 介科伟. 求解带时间窗动态车辆路径问题的改进蚁群算法. *大连理工大学学报*, 2018, 58(5): 539–546. [doi: [10.7511/dllgxb201805015](https://doi.org/10.7511/dllgxb201805015)]
- 宋燕子. 基于模拟退火算法的启发式算法在 VRP 中的应用 [硕士学位论文]. 武汉: 华中师范大学, 2013.
- 苏欣欣, 秦虎, 王恺. 禁忌搜索算法求解带时间窗和多配送员的车辆路径问题. *重庆师范大学学报(自然科学版)*, 2020, 37(1): 22–30.
- Lin S. Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, 1965, 44(10): 2245–2269. [doi: [10.1002/j.1538-7305.1965.tb04146.x](https://doi.org/10.1002/j.1538-7305.1965.tb04146.x)]
- Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 1964, 12(4): 568–581. [doi: [10.1287/opre.12.4.568](https://doi.org/10.1287/opre.12.4.568)]
- Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 1987, 35(2): 254–265. [doi: [10.1287/opre.35.2.254](https://doi.org/10.1287/opre.35.2.254)]
- Gillett BE, Miller LR. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 1974, 22(2): 340–349. [doi: [10.1287/opre.22.2.340](https://doi.org/10.1287/opre.22.2.340)]
- Mole RH, Jameson SR. A sequential route-building algorithm employing a generalised savings criterion. *Journal of the Operational Research Society*, 1976, 27(2): 503–511. [doi: [10.1057/jors.1976.95](https://doi.org/10.1057/jors.1976.95)]
- Woch M, Łebkowski P. Sequential simulated annealing for the vehicle routing problem with time windows. *Decision Making in Manufacturing and Services*, 2009, 3(1–2): 87–100.

(校对责编: 孙君艳)