

# 在线动态订单需求车辆路径规划

李二超,张智钊\*

(兰州理工大学电气工程与信息工程学院,甘肃兰州730050)

**摘要:**针对客户满意度和时间窗时域单一的问题,提出一种多时域分级方式衡量车辆配送进度,该设计细化普通时间窗时域,分成多个时域衡量车辆行进位置,算法方面,遗传算法(genetic algorithm, GA)与变邻域下降搜索算法(variable neighborhood descent, VND)的组合优化形式得到静态预优化路径最优车辆行进线路。动态调度周期中,现有贪婪订单插入算法(greedy order insertion algorithm, GOIA)搜索效率不高,提出一种改进后的贪婪订单插入算法(improved new greedy insertion algorithm, IGOIA),摒弃了GOIA随机插入路径的方式,最有原则的将订单插入到配送路径中去,将其与变邻域下降搜索算法组合优化(Genetic algorithm-Variable Neighborhood Descent, GAVND),对未服务的客户点进行局部优化。通过数学模型优化和求解算法改进,在统一平台上与IGOIA-GAVND、GOIA-GAVND与GOIA-GA2-opt的遗传算法改进形式进行对比试验,改进后的动态订单插入算法在不同规模的Solomon算例下,平均目标值降低了11%,算法平均计算时间降低了2.74 s,实例分析中,成本解分别节约了23%、31%、21%,研究结果证明了原则订单插入算法在滚动周期策略作用下可以获得较高质量的解。

**关键词:**多域分级时间窗;遗传算法;变邻域下降搜索算法;原则订单插入算法;滚动周期

**中图分类号:**U116.2;TP391.9 **文献标志码:**A

**引用格式:**李二超,张智钊.在线动态订单需求车辆路径规划[J].山东大学学报(工学版),2024,54(5):62-73.

LI Erchao, ZHANG Zhizhao. Online dynamic demand vehicle routing planning[J]. Journal of Shandong University (Engineering Science), 2024, 54(5):62-73.

## Online dynamic demand vehicle routing planning

LI Erchao, ZHANG Zhizhao\*

(College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, Gansu, China)

**Abstract:** Aiming at the single problem of customer satisfaction and time window time domain, a multi-time domain hierarchical method was proposed to measure vehicle distribution progress. The design refines the common time window time domain into multiple time domains to measure vehicle traveling position. The combination optimization form of GA and variable neighborhood descent (VND) algorithm was used to obtain the static pre-optimization path optimal vehicle route. In the dynamic scheduling cycle, the existing greedy order insertion algorithm (GOIA) had low search efficiency. An improved new greedy insertion algorithm (IGOIA) was proposed, which abandoned the random insertion path of GOIA and inserts orders into the delivery path in the most principled way. The algorithm was combined with Genetic algorithm-variable Neighborhood Descent (GAVND), and local optimization was performed on unserved customer points. The improved genetic algorithms of IGOIA-GAVND, GOIA-GAVND and GOIA-GA2-opt were compared on a unified platform, the comparison test with other improved forms of genetic algorithm was carried out on the unified platform. The average target value of the improved dynamic order insertion algorithm was reduced by 11% and the average calculation time of the algorithm was reduced by 2.74 s under different scale Solomon examples. The cost solutions save 23%, 31% and 21%, respectively. The research results proved that the principle order insertion algorithm could obtain higher quality solutions under the rolling cycle strategy.

**Keywords:** multi-domain time window; genetic algorithm; variable neighborhood descent search algorithm; improved greedy order insertion algorithm; rolling period

收稿日期:2023-06-21

基金项目:国家自然科学基金资助项目(62063019)

第一作者简介:李二超(1980—),男,教授,博士生导师,博士,主要研究方向为人工智能、多目标优化、机器人控制。E-mail:lecstarr@163.com

\*通信作者简介:张智钊(1997—),男,硕士研究生,主要研究方向为在线动态订单需求车辆路径规划。E-mail:lecstarr@163.com.

## 0 引言

车辆路径规划问题( vehicle routing planning, VRP)被称为离线静态车辆路径问题( static vehicle routing problem, SVRP),在线周期部分则被称为动态在线过程,这一过程中,当新订单产生时,要参考上一周期的车辆路径信息,判断哪些车辆还未完成配送,哪些车辆已经完成配送任务回到了中心车厂,最后通过算法策略决定订单在哪些位置具有可插性,保留最优解为最终插入位置。

国内外专家对动态车辆路径问题做了深入的研究,动态车辆路径规划问题( dynamic vehicle routing planning, DVRP)是指车辆在执行配送任务时,随机接收新增订单需求,自适应处理新订单 VRP 问题。DVRP 问题与现实场景相似,国外很早就对这类问题做了大量研究。文献[1]首次提出 DVRP 概念,主要阐述了动态 VRP 和静态 VRP 的区分,为学者们后续研究指明了方向。文献[2]主要将 DVRP 分成两部分:一部分是事件处理系统,用来管理订单;另一部分是使用变邻域算法和粒子群算法结合,用变邻域算法求解静态 VRP,用改进粒子群算法求解 DVRP。文献[3]使用蚁群算法并采取并行求解机制构造初始路径,运用插入算法减少使用车辆数。文献[4]对 VRP 问题进行编码,建立遗传算法来求解车辆路径规划问题。文献[5]针对带有能力约束的 VRP 问题,在车辆数目固定情况下,通过对 3-opt 算法与遗传算法进行结合,改进遗传算法解决 VRP 问题。文献[6]针对 VRP 问题染色体进行构建,构成单亲遗传算法,通过改进的算法求解 VRP 问题,针对研究效果不错,但算法不具有普遍性,用于其他模型导致收敛快,易陷入局部最优。文献[7]在求解需求可拆分车辆路径规划问题时,通过提出三阶段禁忌算法进行求解,通过对比,改进后的算法相比于其他算法能够有效求得需求可拆分车辆路径规划问题。缺点是基于禁忌算法缺乏全局开发能力,仅限于搜索少量数据量,搜索结果过度依赖初始解。文献[8]研究了关于物流配送和废旧产品回收,将模拟退火算法和邻域搜索算法进行结合,模拟退火算法缺点明显,受温度冷热的影响,冷却速率慢,搜索时间较长,邻域搜索算法本身就需要较长的搜索时间,两者结合在处理静态 VRP 问题时,具有不错的效果,在处理 DVRP 问题时,明显效率低下,时间复杂度较高。相比以上

研究人员现存的研究思路和算法优缺点,针对动态问题对时间效率和解的质量两方面需要,本研究将基于遗传算法进行改进,遗传算法因进化特性和良好的鲁棒性对研究问题没有过多的限制而广泛用于 VRP 问题,进化算子各态历经性使得遗传算法能够进行概率意义的全局搜索,对 NP 问题可以提供极大的灵活性来与其他算法混合研究,因其算法效率较高,可以与其余邻域搜索算法进行组合优化,保证优化效果。

DVRP 已经有了较为深入的探究,但带时间窗的动态车辆路径规划问题( dynamic vehicle routing planning with time window, DVRPTW)中,大多数时间窗研究依然使用的是传统的软时间窗或硬时间窗,每一个客户点都有配送时间范围,这个时域范围被称为时间窗,传统的时间窗时域范围较为单一,车辆每到达一个客户点,可能未在时间窗内,早到或者晚到,需要等待,车辆在这一段时间行驶具体信息模糊不清,只知道车辆最终送达目的地,完成了配送,具体早到多久,迟到多久都含糊不清,传统时间窗针对车辆早到 10 min 和早到 15 min,惩罚权重不能有效衡量车辆行进过程中违反时间窗时确切的惩罚力度,为企业评估优化物流配送成本带来了困难。为了解决这个问题,本研究在原有单时域时间窗基础上,改进为多时域分级时间窗。算法解决 DVRPTW 时,处理小规模数据问题多周期所用的算法为同一种算法<sup>[5]</sup>,主要是以启发式智能算法为主,但这一类算法有弊端,在保证效果的同时,无法保证算法效率,当数据规模增大的时候,算法计算速度呈指数上升,在出现新客户点时,无法保证算法更新后的线路可以继续使用。现实情况中,路况复杂且多变,还有一种是针对动态订单插入改进策略,如贪婪订单插入算法( greedy order insertion algorithm, GOIA)以及 Solomon 提出的最佳客户插入订单插入算法( push forward insertion heuristic, PFIH),这一类算法可以大大增强算法处理模型时的快速性,但配送成本无法得到保证<sup>[9]</sup>。根据现有研究工作,为了使在线动态订单物流问题得到更好解决,除了上述对模型进行优化外,在静态预优化阶段和动态调度阶段都是基于遗传算法进行组合优化。

## 1 数学模型建模描述

动态需求过程分为离线和在线两大周期,离线

周期是静态过程,在这一过程中客户服务地址、需求量及服务时间窗等信息在路径规划前已知。这种条件下,车辆行驶路径一经确定,在后续配送过程中线路固定,这类 DVRPTW 是 VRPTW 扩展问题,以超市配送的生鲜货物为例,取送货为一体的经典 DVRPTW 问题<sup>[10]</sup>,主要是理论模型和算法优化两部分,理论模型根据实际情况,分为订单地址变更处理、订单需求量变更处理以及订单时间窗变更处理等,DVRPTW 问题是多周期的组合<sup>[11]</sup>,多周期过程不作为研究重点。为方便研究动态阶段改进算

法的效果,只将整个过程分为两个大周期,静态订单需求作为第一周期,优先进行调度,作为滚动周期初始输入值,算法计算时间可忽略不计,在形成固定的车辆路径信息后,第二周期建立在一个动态顾客需求集合的基础上调度,通过改进贪婪订单插入算法将订单插入到合理位置,再通过 GAVND 进行局部搜索,完成此次周期的调度<sup>[12]</sup>。

### 1.1 模型约束条件

模型通过数学方式模拟现实车辆配送情况,具体模型参数及其变量如表 1 所示。

表 1 模型参数及变量  
Table 1 Model parameters and variables

符号	定义
$O$	配送中心
$N = \{i\}$	客户点集合,其中 $i=1,2,3,\dots,n$
$K = \{j\}$	配送车辆集合,其中 $j=1,2,3,\dots,n$
$t_i$	客户点 $i$ 的服务时间,其中 $i=1,2,3,\dots,n$
$W_j$	第 $j$ 辆配送车的载重量,其中 $j=1,2,3,\dots,n$
$P_j$	第 $j$ 辆配送车的固定成本,其中 $j=1,2,3,\dots,n$
$c$	配送车辆的油耗成本
$f$	配送车辆制冷剂单价
$G$	制冷剂单位时间消耗
$t_{ij}$	动态订单产生时,车辆在点 $i$ 和 $j$ 之间行驶的时间
$a_1$	单位运输时间内所有车辆产生的能耗成本
$T$	动态订单出现时,车辆已走过的时间
$d_{ij}$	客户点 $i$ 到客户点 $j$ 的路径长度,其中 $i,j \in N$
$t_{ijk}$	第 $K$ 辆车配送客户点 $i$ 到客户点 $j$ 的路径的时间,其中 $i,j \in N$
$T_i^e, T_i^l$	客户点 $i$ 满意的最佳时间窗界限( $e$ 和 $l$ 分别是 early 和 later 的缩写)
$\min T_i^e, \max T_i^l$	客户点 $i$ 能接受的最大时间窗界限
Depart $T$	动态订单需求产生的时间
$T_{si}$	$i$ 客户点开始服务的时刻, $i=1,2,3,\dots,n$
$q_i$	$i$ 点货物量, $i=1,2,3,\dots,n$
$M$	车辆限载的最大载重量
$T_w^{bs}$	表示配送车辆为客户 $w$ 开始服务的时刻
$T_j^{bs*}$	表示插入客户 $w$ 后配送车辆为客户 $j$ 开始服务的时刻

具体约束条件:

(1) 一个客户点的配送服务只能由一辆车配送完成;

(2) 每辆车均从唯一的中心车厂出发,且中心车厂车辆充足;

(3) 配送车辆不得中途为其余客户服务,只能按照规定路线行进;

(4) 中心车厂车辆数设为无限多,车辆均采用同一种车型,用同一个最大车辆载重;

(5) 单辆车配送最大限重大于单个配送客户点的货物需求量,可以多个客户点使用同一辆车进行多点配送;

(6) 配送总费用只计算目标函数所规定的所有成本,复杂启动车辆造成油耗成本等难以估量复杂因素不做研究;

(7) 要求配送车辆在一天时间范围内送完,不考虑疲劳驾驶。

### 1.2 数学模型

数学模型作为一种配送车辆运送的模拟状态,可以使研究人员用极低的成本测试算法的有效性,模型越复杂,约束量越多,算法实现难度越大<sup>[13]</sup>,在本研究除了对时间窗做了部分改进外,其余采用简单模型约束方式,数学模型公式如下:

$$\min z = \sum_{j=N} P_j + \sum_{i=N} \sum_{p=M} \sum_{j=M} cX_{ijk}d_{ip} + \sum_{i=N} \sum_{p=M} \sum_{j=M} fGt_{ijk} + s, \quad (1)$$

式中, $z$ 表示的是求解目标函数, $c$ 为车辆单位能耗, $f$ 为制冷成本, $P$ 为调度车辆所使用的成本, $s$ 为客户满意度,

$$x_{ijk} \begin{cases} 1(\text{车辆 } k \text{ 从点 } i \text{ 行驶到点 } j) \\ 0(\text{否则}) \end{cases}, \quad (2)$$

$$y_{ik} \begin{cases} 1(\text{点 } i \text{ 的任务由车辆 } k \text{ 完成}) \\ 0(\text{否则}) \end{cases}, \quad (3)$$

式(2)(3)中点  $i$  的客户任务只能由一辆车  $k$  去完成,车辆从中心车厂出发,最终回到中心车厂;

$$\sum_{i=1}^N q_i y_{ki} \leq C, \quad (4)$$

$$\begin{cases} \sum_{i=1}^N x_{ij} = y_{kj} (y_{kj} \in \{0,1\}) \\ \sum_{j=1}^N x_{ijk} = y_{kj} (x_{ijk} \in \{0,1\}) \end{cases}, \quad (5)$$

式(4)(5)表示各点货物量由车辆  $k$  完成,单个任务点货物量不得超过车辆总载重限制,并且保证每一个客户点都可以被遍历到;

$$m \leq M, M \in \infty, \quad (6)$$

式(6)中,单个客户点货物量小于车辆货物量载重, $m$ 是单个客户点载质量, $M$ 是单趟车辆满载货物量;

$$u_i(t_i^k) = \begin{cases} f_i = \frac{t_i^k - \min T_i^c}{T_i^c - \min T_i^c}, & t_i^k \in [\min T_i^c, T_i^c] \\ 1, & t_i^k \in [T_i^c, T_i^1] \\ g_i = \frac{\min T_i^1 - t_i^k}{\min T_i^1 - T_i^c}, & t_i^k \in [T_i^1, \min T_i^1] \\ 0, & t_i^k \notin [\min T_i^c, \min T_i^1] \end{cases}, \quad (7)$$

式(7)是多域时间窗的限制,多时域分级时间窗分为最满意时域和可接受时域两部分,这里的时域不是时间窗,而是满意时间窗和可接受时间窗之间的时域差,两种时间窗是包含关系,可接受时间窗包含满意时间窗,传统的单一时间窗只有一个时域,时域之外的惩罚系数只有一个,改进后的时间窗模型更好衡量车辆到达下一个客户点时的具体时间信息,通过惩罚系数分级表示客户满意度<sup>[14]</sup>,当车辆在规定时间内到达的客户点越多时,相应的目标函数值越低,为了更好反应满意度,将一个客户点的满意时间窗设置为  $[T_i^c, T_i^1]$ ,车辆在此范围送达时,满意度设为决策变量 1,因为满意时间窗范围在

可接受时间窗范围内,所以两个时间窗之间有两个间隔时域产生,这两个时域范围分别是  $[\min T_i^c, T_i^c]$  和  $[T_i^1, \max T_i^1]$ 。 $f$  作为一个权重系数用于计算这个间隔时域范围内车辆受到的惩罚,超过可接受时间窗范围  $[\min T_i^c, \max T_i^1]$ ,客户满意度为 0;

$$\sum_{k \in K} \sum_{i \in N} (c_3 \max\{(T_i^c - at_i^k), 0\} + c_4 \max\{(at_i^k - T_i^1), 0\}), \quad (8)$$

$$at_j^k \geq \max\{\min T_j^c, x_j^k(at_j^k + wt_i^k + st_i^k + t_{ij})\} \quad \forall i \in I_s(t) \cup I_k(t) \cup I_o, \quad (9)$$

式(8)(9)公式为多域时间窗约束条件,车辆行驶过程假设匀速行驶,其中的  $at_i^k$  表示车辆  $k$  到达客户节点  $i$  的时间,式中  $T_i^c - at_i^k$  得到的差值即为车辆服务客户点  $i$  早到的时间, $K$  和  $N$  分别为车辆集合和客户点集合, $c_3$  为车辆在客户  $i$  期望时间前到达的惩罚权重系数, $c_4$  为车辆在客户  $i$  期望时间后到达的惩罚权重系数,通过每个时域赋权重系数,有效的分级表示车辆在不同时域范围内行驶状态;

$$at_j^k \leq MLT_j, \quad \forall j \in I_o \cup I_s(t), \quad \forall k \in K, \quad (10)$$

$$\begin{cases} \sum_{i=1}^N x_{iok} = 1 \\ \sum_{j=1}^N x_{ojk} = 1 \\ \sum_{\substack{i \in U_k \\ j \in U_k}} x_{ijk} = 1 \end{cases} \quad U_k = \{i, X_{ki} = 1\}, \quad (11)$$

式(10)(11)为客户满意度的约束条件。

### 1.3 滚动周期策略设计

动态的研究策略主要采用滚动周期<sup>[15]</sup>策略方式处理动态订单的产生,本研究采用虚拟点方式解决滚动周期的问题,假设选择路径  $1 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 2 \rightarrow 6 \rightarrow 1$ ,中心车厂以及各客户点编码方式用数字编码设计,从中心车厂出发再回到中心车厂,形成一条完整的闭环回路,除中心车厂外,在产生动态订单时,需要考虑上一周期车辆信息,判断车辆是否已经完成上一周期的配送,回到了中心车厂,如果上一周期有还未完成配送的车辆,则需要判断目前车辆行进位置,建立未配送车辆客户集。

$$\min T_i^1 \leq T_{ij} + T \cup \max T_i^c > T_{ij} + T, \quad (12)$$

$$T_{ki-1} \leq T_{ij} + T \cup T_{ki} > T_{ij} + T, \quad (13)$$

式(12)(13)用于判断当前车辆行驶具体位置,是否满足时间窗限制和服务时间限制,如果满足,则在点  $i$  和  $j$  之间做一个虚拟点,这个虚拟点通过计算得到相应的坐标,再将虚拟点放入未配送客户点集中,如图 1 所示。

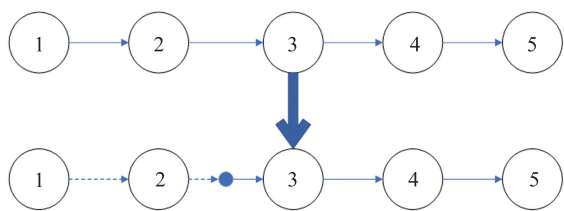


图1 虚拟点设置图

Fig.1 Virtual point Settings diagram

车辆行进至编号2和编号3之间,计算出此虚拟点的位置坐标,将虚拟点并入未服务集合中,这里的未服务集合是一个临时承载未服务客户点数据的数组,将此位置作为可行解的起始点,当前货物量和时间窗为

$$q_{n+k} = G_k, \quad (14)$$

$$\min T_{i+k}^e = \max T_{i+k}^l = t_i. \quad (15)$$

式(14)中  $G_k$  为车辆  $k$  在已走路程货物量的总和,式(15)为当前虚拟点时间窗区域,车辆在虚拟点时,时间窗区域缩小为一个点,  $t_i$  是车辆到达虚拟点的时间。通过虚拟点设置可以将当前每辆车的行进位置标出,将动态VRP问题通过滚动周期分为多周期的静态问题。

## 2 优化算法设计

搜索算法都是基于遗传算法进行优化,改进方式不同在于预优化采用策略是将VND嵌入到遗传算法的变异之后,增大邻域搜索范围,动态调度算法针对GOIA算法进行改进,在插入订单之后使用遗传算法与变邻域下降搜索算法组合优化形式(improved greedy order insertion algorithm—genetic algorithm and variable neighborhood, IGOIA—GAVND)进行深度搜索,依次完成迭代。对算法优化改进后,增强响应速度和有效降低成本。

GAVND组合优化形式对静态问题进行初始调度,将静态阶段看成动态阶段第一个周期,在完成第一阶段调度,会出现一组完整固定线路以及车辆信息,在出现新增动态订单时,设置虚拟点方式分为两大静态周期<sup>[9]</sup>,插入算法将新的订单插入到合理的位置,最终再用GAVND进行局部优化,为了方便研究算法效果,在静态和动态阶段均采用固定输入信息。

### 2.1 GAVND算法设计

GAVND是遗传算法与深度变邻域下降搜索算法的组合优化,静态部分VRP问题,遗传算法参数设定为种群Popsiz为50,最大迭代数IterMax为100,交叉概率Pc为0.8,变异概率Pm为0.1,优先

对VRP问题的可行解进行编码并产生初始种群,对初始种群根据数学模型的输入条件,计算适应度并对种群进行选择,根据目标函数终止条件为(1)当前最优解连续10代不发生变化;(2)当迭代步数超过100步。

#### 2.1.1 编码操作

已有的对车辆路径规划的编码操作形式多样,其中一种是将所有客户点用数字表示,在算法优化后形成一条可行解后,基于这条可行解随机设置固定点,这里固定点在编码操作中设置为中心车厂,一般用固定数字“1”或“0”,将这条可行解分为若干条路径,每一条车辆路径都从中心车厂出发并回到中心车厂。这一类编码有一个弊端,容易忽略车辆容量,当遇到车辆容量有限制的模型,很难有效的满足需求,第二种是编码方式,采用“先排序,后聚类”的方法<sup>[16]</sup>,染色体编码为包含全部客户点编号的一个不重复排序,路径分割满足总货物需求量不超过该车载重,即一个染色体中,按从左到右顺序将所有客户点数字排成一条线,这种编码方式占用储存量相对较小,方案会覆盖到所有配货点,编码效率高,假设最优染色体配送线路为4→7→3→8→9→6→10→2→5,车辆容量在配送到客户点“8”时,满载货物质量C清零,此时这辆车需要返回中心车厂,第一辆车的路径信息为4→7→3→8。

#### 2.1.2 GAVND算法步骤

求解模型的目标函数的倒数定为适应度函数  $F$ ,再对遗传算法进行选择,交叉和变异算子的选择。

##### (1) 选择算子

对种群中所有可行解排序,要选择出适应度函数最大,即目标函数最小,选择算子有5种选择函数,采用轮盘赌策略进行选择。计算概率为:

$$P(x_i) = \frac{F(x_i)}{\sum_{j=1}^N F(x_j)}, \quad (16)$$

$$L_i = \sum_{j=1}^i P(x_j), \quad (17)$$

式中,  $P(x_i)$  是每一个可行解被遗传到下一代种群中的概率,  $L_i$  为每个个体的累积概率。

##### (2) 交叉算子

初始产生  $[0, 1]$  区间内随机整数  $r_1$  和  $r_2$  确定子代和父代交叉位置,对两个位置的中间数据进行交叉,假设两个随机数  $r_1$  和  $r_2$  分别等于4和7,两条染色体为9→5→1→3→7→4→2→10→8→6和10→5→1→6→3→8→7→2→1→9根据随机产生的

交叉位置,将父代染色体的1和10中间的客户点和子代染色体4和2中间的客户点交叉互换,交叉后同一个个体中有重复客户点采用部分映射,结果为9→5→1→6→3→8→7→10→4→2与10→5→8→3→7→4→2→6→1→9。

(3) 变异操作

变异策略采取随机选取两点将其位置互换,假设产生[1,10]内的随机整数 $r_1$ 和 $r_2$ 分别为4和7,确定两个位置,进行互换,染色体为车辆路9→5→1→6→3→8→7→10→4→2,染色体形态为9→5→1→7→3→8→6→10→4→2。

(4) 重插入子代

重插入策略就是将变异交叉后个体重新补回种群中。获得基本遗传算法在此次迭代中的最优解。

(5) 变邻域下降搜索更新解

变邻域下降搜索更新解,邻域在不同问题中所指代的含义也不同,是一种较为成熟的改进型局部搜索算法,此算法的主要思想是采用多个不同的邻域进行系统搜索,当所在的当前邻域无法改进解时,则切换到另一个邻域改进解的质量,当前邻域搜索改进解质量,在此邻域继续搜索,将变邻域下降搜索算法嵌入到遗传算法中,每一次遗传算法迭代均会产生一个新的个体,利用VND算法对个体路径进行局部搜索,邻域搜索算法设定 $N_k$ 个邻域结构,邻域结构为 $N_k = N_1, N_2, \dots, N_n$ ,这里的 $N$ 设定为Exchange优化邻域、Insert优化邻域以及2-opt优化邻域,选择一条车辆路径作为初始邻域,从Exchange优化邻域开始优化,优化邻域的互相切换被称为邻域动作;若在此邻域就找到了改进解,则继续在此邻域继续扰动;若没找到改进解,到下一个邻域重复操作。本研究采用路径内优化方法,3种径内优化邻域的搜索效果和范围一样<sup>[11]</sup>,都是随机选择一条车辆路径,在此车辆路径内进行邻域操作,如图(2)~(4)所示。

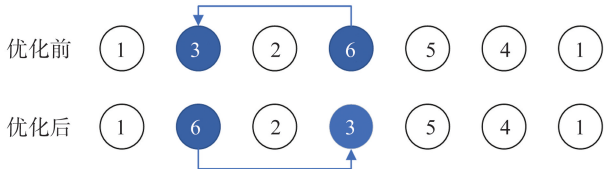


图2 Exchange 优化  
Fig.2 Exchange optimization

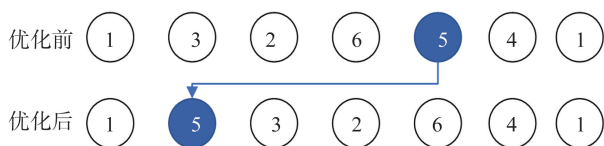


图3 Insert 优化  
Fig.3 Insert optimization

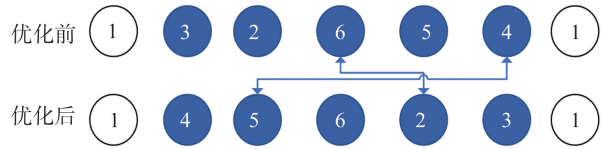


图4 2-opt 优化  
Fig.4 2-opt optimization

① Exchange 优化

选择一条路径,交换路径中两个位置,最终形成一条新的路径。

② Insert 优化

选择一条路径,将路径中一个节点插入此路径的其他位置。

③ 2-opt 优化

选择一条路径,将路径中各节点逆序遍历访问,生成一条新的路径。

2.2 原则插入算法设计

融入PFIH算法策略与贪婪插入算法思想进行改进,插入算法解决订单插入的问题,在解决动态订单问题时,侧重两种:(1)是全局重新优化策略。(2)是局部优化策略。全局重新优化主要使用启发式算法,这类算法在解决小规模订单问题上具有很好效果,在处理大规模问题上,耗费的时间多,很难满足现实复杂环境下运输车辆配送的时效性和降低配送成本两方面需要,当数据规模增大的时候,算法计算速度呈指数上升,出现在新增客户点时,无法保证更新后的当前线路可以继续使用。相较于全局优化策略局部优化策略,可以节约时间成本,只针对未配送的客户点进行调度。本研究采用局部优化策略,利用动态订单插入算法进行组合优化,基于贪婪订单插入算法改进,贪婪订单插入算法将订单插入到客户点之间,寻找订单插入位置时,随机选择车辆路径进行插入,导致算法具有偶然性,稳定性较差。这种随机方式有效地提升了算法效率,很难保证解的质量,基于贪婪订单插入算法进行改进,提出原则订单插入算法(Improved greedy order insertion algorithm, IGOIA),IGOIA算法将Solomon的PFIH算法思想融入到贪婪订单插入算法中,将新客户点插入*i*和*j*客户点之间,原则订单插入算法的算法步骤如下。

步骤1: 车辆执行初始配送方案,并确定最大迭代次数 Gen\_max。

步骤2: 在 departT 时刻,更新在途车辆状态信息,建立未服务客户点集合 *B*。

步骤3: 将任务依次最有原则的插入到未服务

客户集合中,假设第一个随机客户为  $w$ ,按顺序遍历上一周期在途的车辆信息,将  $w$  插入到配送路径两个相邻客户点  $i$  和  $j$  之间。判断客户  $w$ 、 $j$  的服务时间窗是否得到满足,同时新增客户点以及的货物量是否小于车辆最大满载货物量:

$$T_w^{bs} \leq \min T_w^l, \quad (18)$$

$$T_j^{bs*} = T_w^{bs} + T_w^s + T_{wj} \leq \max T_{w/j}^e, \quad (19)$$

$$q_i \leq C, \quad (20)$$

式中,  $T_w^{bs}$  表示配送车辆为客户  $w$  开始服务的时刻,  $T_j^{bs*}$  表示插入客户  $w$  后配送车辆为客户  $j$  开始服务的时刻,  $\min T_w^e$ 、 $\max T_j^l$  表示客户  $w$  和  $j$  的最早和最晚服务时间窗,  $T_w^s$  表示客户  $w$  需要服务的时间,  $T_{wj}$  表示客户  $w$ 、 $j$  之间的行驶时间,  $q_i$  表示各点货物量,  $C$  表示车载最大货物量;

步骤4:当满足步骤3的条件,则判断客户  $w$  具有可插性, best  $F$  的适应度:

$$\frac{1}{\sum_{j \in M} p_j + \sum_{i \in N} \sum_{p \in M} \sum_{j \in M} cX_{ijk} d_{ip} + \sum_{i \in N} \sum_{p \in M} \sum_{j \in M} fGt_{ijk} + s} \quad (21)$$

步骤5:将插入的需求客户点依次遍历在途车辆信息,当新增需求客户点插入原路径,选择 best  $F$  最优的插入位置,作为此随机客户点的最终插入位置,如在途所有车辆信息均不满足步骤4条件,则再调一辆车完成配送任务。

步骤6:插入点全部插入结束,利用 GAVND 进行局部优化,结束本周期的调度。

### 3 试验仿真及算例分析

将试验在搭载 3.3 GHz, AMD Ryzen5 处理器和 16 G 内存的计算机平台上实现,离线静态部分使用 GAVND 进行搜索调度形成完整的路径信息,在线动态部分用 5 组算法用 Solomon 数据集在 MATLAB 2016 进行测试对比试验<sup>[7]</sup>,离线静态输入信息包括 1 个中心车厂和 24 个客户节点,客户节点随机分布,客户需求量固定不具有规律性,车辆最大质量  $C$  为 200 kg,中心车厂坐标为 [40,50],时间窗早到惩罚为 5 元/min,配送车辆单位油耗成本  $c$  为 3 元,配送车辆制冷剂单价  $f$  为 0.2 元,制冷剂单位时间消耗  $G$  为 2 元,在线动态部分依旧采用固定但不具有规律性的信息输入,其中包括新增客户点的增加、上一周期正在配送过程中客户点的坐标变化、配送需求量变化以及客户点删减。

#### 3.1 离线周期部分

离线周期的初始化车辆路径规划的客户需求信息如表 2 所示,先对配送中心调度优化,模拟线路中,采用欧式距离计算两个客户点间的距离,利用 GAVND 算法对初始的路径进行优化,得到具体的线路和相关车辆具体信息,滚动周期动态优化调度方案,如图 5 所示。

车辆路径信息如表 3 所列,离线周期配送路线如图 6 所示。

表 2 初始车辆路径信息  
Table 2 Initial vehicle routing information

节点编号	坐标	服务量	可接受时间窗	满意时间窗	服务时间
1	(41,49)	10	[0,204]	[10,194]	10
2	(35,17)	7	[0,202]	[10,192]	10
3	(55,45)	13	[0,197]	[10,187]	10
4	(55,20)	19	[149,159]	[149,159]	10
5	(15,30)	26	[0,199]	[10,189]	10
6	(25,30)	3	[0,208]	[10,198]	10
7	(20,50)	5	[0,198]	[10,188]	10
8	(10,43)	9	[95,105]	[85,105]	10
9	(55,60)	16	[97,107]	[87,107]	10
10	(30,60)	16	[0,194]	[10,184]	10
11	(20,65)	12	[67,77]	[67,77]	10
12	(50,35)	19	[0,205]	[10,195]	10
13	(20,65)	23	[15,16]	[169,169]	10
14	(15,10)	20	[0,187]	[10,177]	10
15	(30,5)	8	[61,71]	[61,71]	10
16	(10,20)	19	[0,190]	[10,180]	10
17	(5,30)	2	[0,189]	[10,179]	10

表2(续)

节点编号	坐标	服务量	可接受时间窗	满意时间窗	服务时间
18	(20,40)	12	[0,204]	[10,194]	10
19	(15,60)	2	[0,187]	[10,177]	10
20	(45,65)	12	[0,188]	[10,178]	10
21	(45,20)	17	[0,201]	[10,191]	10
22	(45,10)	18	[0,193]	[10,183]	10
23	(55,5)	29	[68,78]	[78,78]	10
24	(65,35)	3	[0,190]	[10,180]	10
25	(65,20)	6	[172,182]	[182,182]	10

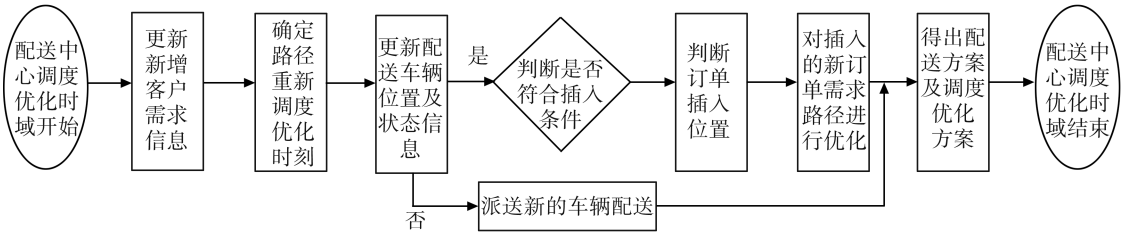


图5 滚动周期动态调度优化方法图

Fig.5 Rolling cycle dynamic scheduling optimization method diagram

表3 车辆路径信息

Table 3 Vehicle routing information

车辆	路径信息	距离/km
1	0 → 5 → 16 → 17 → 8 → 18 → 7 → 0	168.7
2	0 → 24 → 12 → 3 → 9 → 0	128.3
3	0 → 15 → 2 → 6 → 4 → 25 → 0	221.1
4	0 → 1 → 23 → 22 → 14 → 13 → 0	198.0
5	0 → 20 → 19 → 11 → 10 → 0	118.6

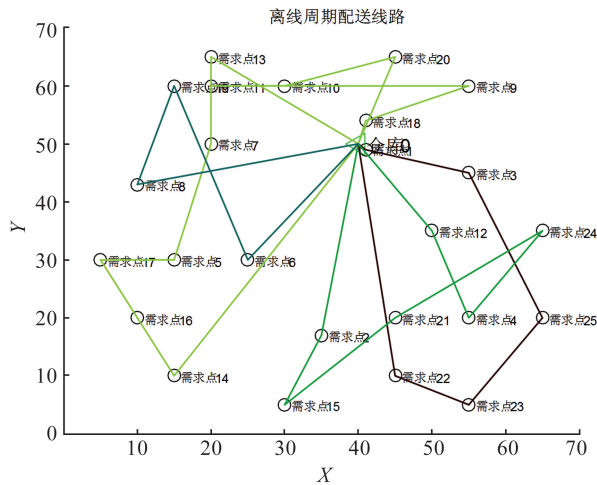


图6 离线周期配送路线图

Fig.6 Off-line cycle distribution roadmap

### 3.2 在线动态算例分析

结合 DVRPTW 的特性,采用的算例是由研究人员广泛使用的国际通用 Solomon 数据集修改而成<sup>[17]</sup>,Solomon 数据集是为了方便研究人员研究,仿照实际需求制作的仿真数据集,包括顾客位置坐标、需求量、时间窗以及服务时间<sup>[18]</sup>。为了验证提出的原则订单插入算法在处理 DVRPTW 问题上的有效性,将原则订单插入算法 IGOIA 在同一个平

台上用 30 个算例进行计算分析,20 组动态订单输入数据分析如表 4 所示,30 组动态订单输入数据分析如表 5 所示,对比的订单插入算法为取送货一体化的动态车辆路径问题中所采用的贪婪订单插入算法 GOIA<sup>[19]</sup>,局部优化效果上,动态局部搜索算法采用的是 GAVND,对比局部优化算法为基于改进遗传算法的同城快递配送模型所使用的 GA2-opt 算法<sup>[20]</sup>。将 IGOIA-GAVND、GOIA-GAVND 与 GOIA-GA2-opt 等 3 组算法分别运行 10 次并取平均值。表 4 中  $L$  代表路径长度,也就是目标函数取值, $t$  表示 CPU 运行的时间。

表4 动态时刻 20 组数据量比较结果

Table 4 Comparison results of 20 groups of data volume at dynamic time

算例	IGOIA-GAVND		GOIA-GAVND		GOIA-GA2-opt	
	$L/km$	$t/s$	$L/km$	$t/s$	$L/km$	$t/s$
c101	40.3	2.05	45.6	5.84	45.3	0.99
c102	43.6	1.92	46.0	5.66	46.0	1.19
c103	45.2	2.74	51.1	7.91	52.2	1.27
c104	46.3	1.74	49.4	5.69	51.1	0.93
c105	67.5	1.74	72.2	4.12	70.4	1.06
c106	67.8	1.81	83.2	3.02	69.9	0.93
c107	64.6	1.93	77.8	4.91	72.5	1.11
c201	69.2	1.89	85.9	5.91	75.9	1.11
c202	72.8	1.79	73.8	2.98	79.8	1.07
c203	59.5	1.75	69.5	4.31	73.2	1.12
c204	71.5	1.81	81.9	3.67	82.1	1.13
c205	80.9	1.93	92.4	2.26	94.5	1.12
c206	73.9	1.97	77.7	6.65	78.1	0.88
c207	76.6	2.08	86.2	4.45	86.0	1.91
c208	61.6	1.77	66.0	2.55	66.6	0.74
平均	62.8	1.92	70.6	4.66	69.6	1.11

表4表示的是在动态周期中,出现20组新增客户信息的情况下各算法在处理结果的对比情况,本研究的IGOIA-GAVND算法在计算速度上明显优于GOIA-GAVND和GOIA-GA2-opt两种算法,相较于GOIA-GAVND算法,IGOIA-GAVND算法平均目标值降低了11%,算法平均计算时间降低了2.74 s,相较于GOIA-GAVND算法,IGOIA-GAVND算法平均计算时间仅增加了0.81 s,但平均目标值降低了10%;从表4的c103~c208算例来看,IGOIA-GAVND算法改进效果明显,算法可行有效,各算法在以上Solomon算例中每次迭代最优解平均值的比较可以看出,原则订单插入算法找到最优插入位置后,局部优化算法GAVND在第一次迭代的时候就找到了最优目标解,如图7所示。为进一步验证算法在处理不同订单需求时的有效性,采用r101~r203进行验证分析,每个算例调用30组改进数据,即动态周期变化30个客户点,对比结果如表5所示,各算法迭代最优解平均值如图7所示。

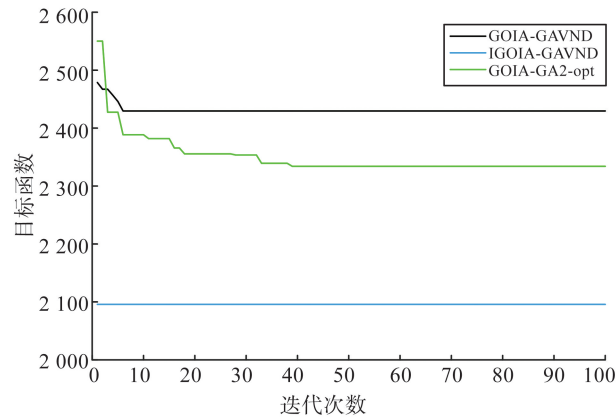


图7 Solomon算例中迭代目标函数比较

Fig.7 Iterative objective function comparison in Solomon example

通过Solomon的r101~r203的数据集比较可以看出,当动态订单数量增加时,30组订单量较20组订单量时,相比GOIA-GAVND算法计算时间平均值增加了12.22 s,改进的IGOIA-GAVND算法计算时间降低了0.07 s,基本保持稳定,没有因为数据量的升高而降低算法效率,3种算法较表4数据量目标值分别增加了22%、18%、25%,成本都有所上升,差距较小。在动态新增30组订单量时,IGOIA-GAVND相较于GOIA-GAVND算法目标值降低了7%,IGOIA-GAVND相较于GOIA-GA2-opt算法目标值降低了14%,GOIA-GAVND和GOIA-GA2-opt使用的贪婪订单插入算法在订单插入操作结束时并未完全找到最优插入位置,局部优化操作起主要的搜索作用,算法时间和最优目标值都未达到最满意的结

果,通过以上算例结果的比较可以明显看出原则订单插入算法得到的目标值均为最优,算法运算时间也相较于贪婪订单插入算法得到了改善。

表5 动态时刻30组数据量比较结果

Table 5 Comparison results of 30 groups of data volume at dynamic time

算例	IGOIA-GAVND		GOIA-GAVND		GOIA-GA2-opt	
	L/km	t/s	L/km	t/s	L/km	t/s
r101	102.0	2.21	109.0	21.46	114.0	1.21
r102	89.0	0.87	90.8	19.39	96.8	1.06
r103	92.4	2.22	97.8	15.61	99.0	1.41
r104	94.3	2.91	96.2	19.12	108.0	1.21
r105	99.1	2.17	100	9.72	109.0	1.22
r106	95.9	2.26	98.4	8.65	115.0	1.53
r107	92.4	2.25	93.0	37.11	103.0	1.42
r108	94.3	2.16	95.0	39.66	106.0	1.41
r109	69.6	1.92	74.8	17.06	93.3	1.12
r110	43.6	0.83	51.7	10.54	63.3	1.21
r111	69.3	2.01	74.3	20.96	76.5	1.31
r112	68.1	2.04	77.1	7.18	75.6	1.13
r201	58.0	1.11	69.4	6.16	77.7	1.15
r202	74.6	1.98	83.3	18.61	80.5	1.12
r203	56.3	0.85	73.0	2.03	66.3	1.06
平均值	80.0	1.85	85.6	16.88	92.3	1.24

## 4 实际案例应用分析

在实际背景下,进一步验证改进算法有效性,本文采用某超市连锁店作为实际案例,测试平台与Solomon数据集测试所使用的平台一致,均采用搭载3.3 GHz, AMD Ryzen5处理器和16G内存的计算机平台上实现。

### 4.1 案例描述

实际案例的坐标包括供应仓库以及各连锁店的具体位置坐标位置,通过高德地图截取连锁店具体位置如图8所示。



图8 连锁店分布位置

Fig.8 Location of chain stores

仓库的坐标为(40,50),为方便测试,部分连锁店坐标投影到XOY平面上,将位置经纬度转化为直角坐标,用GAVND算法优先得到离线车辆配送路径如图9所示。

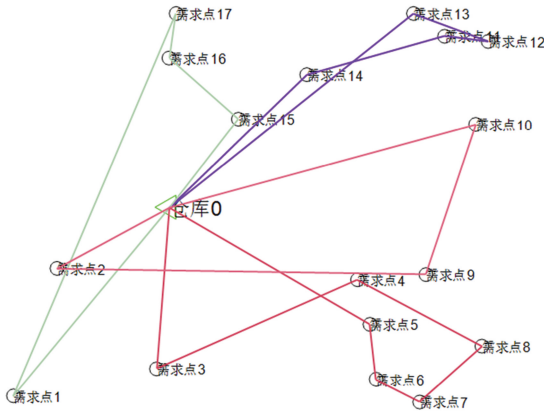


图9 离线车辆路径配送线路  
Fig.9 Off-line vehicle route distribution lines

### 4.2 数据分析

本研究中实际应用案例相较于数据集使用的数据量较少,当数据量较少时,订单插入算法的时效性优势不明显,在实例比对中,采用3种算法进行比较,分别是IGOIA-GAVND、GOIA-GAVND和GA2-opt。IGOIA-GAVND和GOIA-GAVND算法为了测试原则订单插入算法相较于贪婪订单插入算法的优劣性,GA2-opt是改进后的启发式算法,主要应用于小规模订单量的情况,3种算法将从CPU运行时间以及成本两方面进行测试试验,将每种算法运行10次,取最优配送成本、最差配送成本以及平均配送成本,动态时刻的在线订单数据依旧采用规则,不具有普遍性的订单数据,坐标分别是(17,18)、(18,55)、(19,72)、(20,32)、(21,19)、(22,43)、(23,62)、(24,12)。各类算法比对如表6、7所示。

表6 各类动态订单阶段算法配送成本  
Table 6 Various dynamic order stage algorithm delivery costs

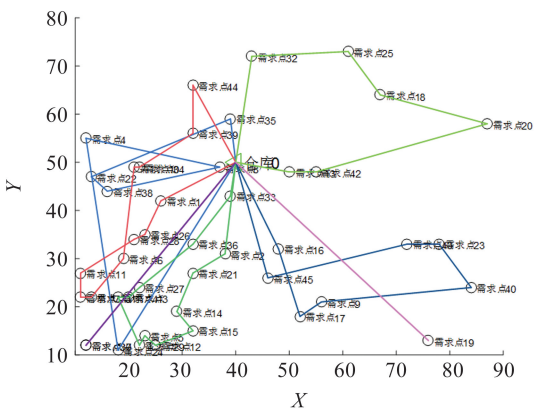
配送成本								
IGOIA-GAVND			GOIA-GAVND			GOIA-GA2-opt		
最佳值	平均值	最差值	最佳值	平均值	最差值	最佳值	平均值	最差值
<b>403.4</b>	<b>410.2</b>	<b>418.7</b>	461.4	455.1	468.8	405.2	426.1	446.2

表7 各类动态订单阶段算法CPU运行时间  
Table 7 Various dynamic order phase algorithm CPU running time

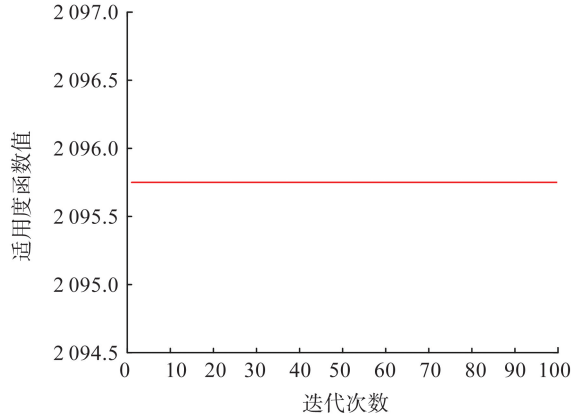
CPU运行时间								
IGOIA-GAVND			GOIA-GAVND			GOIA-GA2-opt		
最佳值	平均值	最差值	最佳值	平均值	最差值	最佳值	平均值	最差值
<b>1.6</b>	<b>1.5</b>	<b>1.8</b>	2.1	2.2	2.3	1.1	1.2	1.5

表6、7中数据黑体数字表示最优解,由表6、7可知:小规模在线动态订单量实际问题中,原则订单插入算法处理在线动态订单时在配送成本方面要优于其余对比的算法,IGOIA-GAVND算法中所使用的变邻域下降搜索算法在订单插入后进行了深度搜索,从CPU运行时间的最佳值、平均值以及最差值三方面来看,IGOIA-GAVND运行时间以上三方面都要略高于GA2-opt,分别高出0.4%、3.7%、6.1%,由数据可以看出IGOIA-GAVND算法运行更加稳定,IGOIA-GAVND相较于GOIA-GAVND配送成本分别节约了23%、31%、21%,通过控制变量法得到IGOIA算法目标解和算法运行时间均要优于GOIA算法,3种算法对应的线路图以及收敛效果如图10~12所示。由图10可以看出,IGOIA-GAVND算法在第一次深度搜索之前就找到了最优订单插入位置以及最优配送路径方

案。由图11可知GA2-opt算法前10次迭代收敛速度快但未达到全局最优,直到第44次迭代找到最优成本解,此算法劣势在于用于在线动态车辆路径规划时,订单量过大易陷入局部最优,同时无法保证算法时效性,在此案例情况下并不存在时效性的平衡处理问题。在线订单的规模小,3种算法的CPU运行时间对比过后,改进的IGOIA-GAVND算法的CPU运行时间相较于GA2-opt较差,测试结果与算例测试结果一致,原因同为增加了邻域搜索导致运行时间较多。由图12可知,相较于对比算法GOIA-GAVND在深度搜索之前并未完全找到最优订单插入位置,算法迭代第8次时,找到了最优的配送路径方案,GOIA的订单插入位置搜索效率低于IGOIA。从算例以及实例进行分析,研究提出的IGOIA-GAVND算法有效。



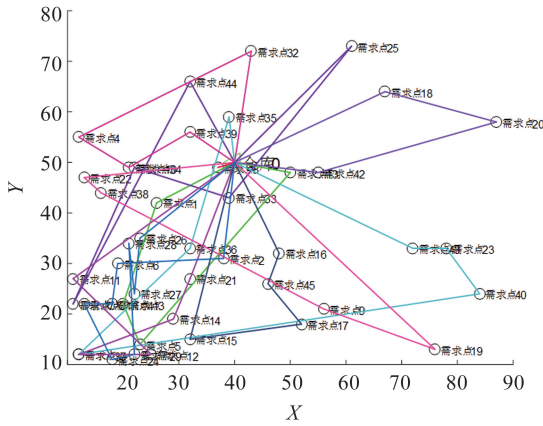
(a) IGOIA-GAVND算法绘制配送方案



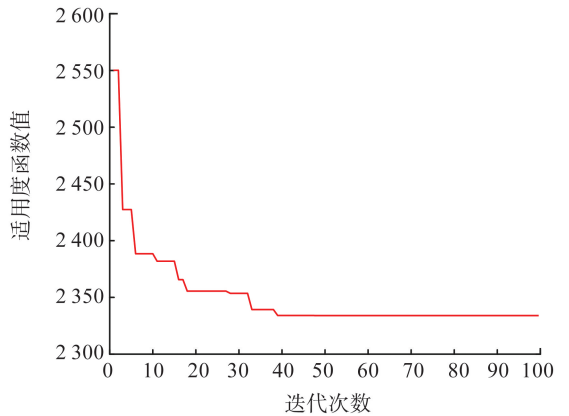
(b) IGOIA-GAVND算法配送成本

图10 IGOIA-GAVND算法配送方案及配送成本

Fig.10 IGOIA-GAVND algorithm distribution scheme and distribution cost



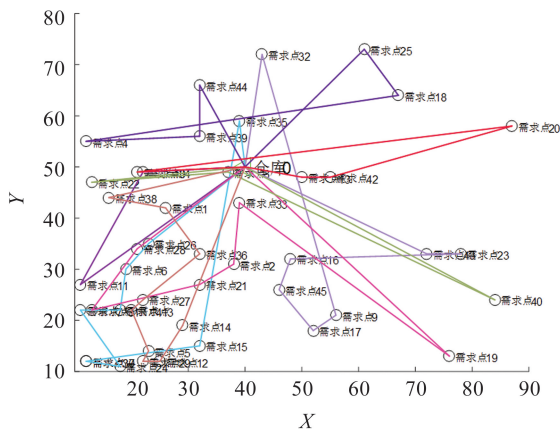
(a) GA2-opt算法绘制配送方案



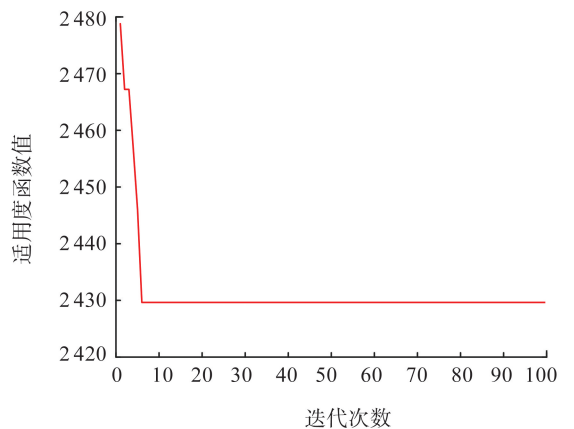
(b) GA2-opt算法配送成本

图11 GA2-opt算法配送方案及配送成本

Fig.11 GA2-opt algorithm distribution scheme and distribution cost



(a) GOIA-GAVND算法绘制配送方案



(b) GOIA-GAVND算法配送成本

图12 GOIA-GAVND算法配送方案及配送成本

Fig.12 GOIA-GAVND algorithm distribution scheme and distribution cost

### 5 结论

在线动态车辆路径问题种类繁多,目前研究人员所研究的问题主要是基于实际要求的理论性研

究,模型上为了有效衡量客户满意度,本研究针对性的对时间窗进行改进,建立多域分级时间窗策略,对车辆到达时间进行分级奖励或惩罚<sup>[21]</sup>,再将其通过加权赋值统一目标函数,在设计优化算法时,本研究针对性的改进了局部优化策略中的订单

插入算法,有效的降低了算法运算时间以及配送成本。利用30组不同规模的Solomon仿真数据测试了算法的效果,与未改进的算法进行对比试验,同时测试了实际背景下某超市实例的应用并进行了相应的分析,通过不同角度的试验分析,本研究提出的算法策略可以有效解决面向动态订单的车辆路径规划问题,不足之处在于算法所处理的模型没有完全贴合实际应用,比如:(1)车辆行驶速度始终假设匀速,(2)车辆路径规划问题中复杂交通问题也没有考虑进去。以上两点将作为后续研究重点之一。

#### 参考文献:

- [1] BOZORGI-AMIRI A, KHORSI M. A dynamic multi-objective location-routing model for relief logistic planning under uncertainty on demand, travel time, and cost parameters [J]. *The International Journal of Advanced Manufacturing Technology*, 2016, 85: 1633-1648.
- [2] JAMES J Q, YU W, GU J. Online vehicle routing with neural combinatorial optimization and deep reinforcement learning[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 20(10): 3806-3817.
- [3] SAINT-GUILLAIN M, PAQUAY C, LIMBOURG S. Time-dependent stochastic vehicle routing problem with random requests: Application to online police patrol management in Brussels[J]. *European Journal of Operational Research*, 2021, 292(3): 869-885.
- [4] 王仁民, 闭应洲, 刘阿宁, 等. 改进变邻域搜索算法求解动态车辆路径问题[J]. *计算机工程与应用*, 2014, 50(2): 237-241.  
WANG Renmin, GUAN Yingzhou, LIU Aning, et al. Improved variable neighborhood search algorithm for Dynamic Vehicle Routing Problem[J]. *Computer Engineering and Applications*, 2014, 50(2): 237-241.
- [5] 马欢, 张建伟, 赵进超, 等. 求解VRPSDP的变邻域混合遗传算法[J]. *郑州大学学报(工学版)*, 2015, 36(3): 120-124.  
MA Huan, ZHANG Jianwei, ZHAO Jinchao, et al. Variable neighborhood hybrid genetic algorithm for VRPSDP [J]. *Journal of Zhengzhou University (Engineering Edition)*, 2015, 36(3): 120-124.
- [6] 康熙沛, 杨家其, 余昊, 等. 基于离散灰狼算法的带软时间窗车辆路径规划问题[J]. *武汉理工大学学报(交通科学与工程版)*, 2022, 46(4): 598-603.  
KANG Xipei, YANG Jiaqi, YU Hao, et al. Vehicle path planning with soft time window based on discrete Gray Wolf algorithm [J]. *Journal of Wuhan University of Technology (Transportation Science and Engineering)*, 2022, 46(4): 598-603.
- [7] SOLOMON M M. Algorithms for the vehicle routing and scheduling problems with time window constraints [J]. *Operations Research*, 1987, 35(2): 254-265.
- [8] 郭富蓉, 巩建忠, 崔袁丁. 时变需求环境下同时取送货的车辆路径问题优化研究[J]. *甘肃科技纵横*, 2021, 50(5): 51-56.  
GUO Furong, GONG Jianzhong, CUI Yuan-Ding. Research on optimization of vehicle routing problem for simultaneous pickup and delivery under time-varying demand environment[J]. *Gansu Science and Technology*, 2021, 50(5): 51-56.
- [9] 陈萍, 黄厚宽, 董兴业. 求解卸装一体化的车辆路径问题的混合启发式算法[J]. *计算机学报*, 2008(4): 565-573.  
CHEN Ping, HUANG Houkuan, DONG Xingye. Hybrid Heuristic Algorithm for Solving Vehicle Routing Problem with integrated unloading and loading[J]. *Chinese Journal of Computers*, 2008(4): 565-573.
- [10] 陈久梅, 李英娟, 胡婷, 等. 开放式带时间窗车辆路径问题及变邻域搜索算法[J]. *计算机集成制造系统*, 2021, 27(10): 3014-3025.  
CHEN Jiumei, LI YingJuan, HU Ting, et al. Open vehicle routing problem with time window and variable neighborhood search algorithm[J]. *Computer Integrated Manufacturing Systems*, 2021, 27(10): 3014-3025.
- [11] 李兵, 郑四发, 曹剑东, 等. 求解客户需求动态变化的车辆路径规划方法[J]. *交通运输工程学报*, 2007(1): 106-110.  
LI Bing, ZHENG Sifa, CAO Jiandong, et al. Vehicle path planning method to Solve the dynamic change of Customer Demand[J]. *Journal of Traffic and Transportation Engineering*, 2007(1): 106-110.
- [12] XUE G, WANG Y, GUAN X, et al. A combined GA-TS algorithm for two-echelon dynamic vehicle routing with proactive satellite stations[J]. *Computers & Industrial Engineering*, 2022, 164(2): 107899.
- [13] FABRI A, RECHT P. On dynamic pickup and delivery vehicle routing with several time windows and waiting times[J]. *Transportation Research Part B: Methodological*, 2006, 40(4): 335-350.
- [14] 宋娟, 崔艳. 基于改进遗传算法的同城快递配送模型[J]. *电子技术应用*, 2014, 40(12): 136-139.  
SONG Juan, CUI Yan. Same-city express distribution model based on improved genetic algorithm [J]. *Application of Electronic Technique*, 2014, 40(12): 136-139.