doi：10. 12068/j. issn. 1005－3026. 2026. 20259006

# 同时取送货场景下的电动车辆路径规划算法

李晓辉，孙炜桐，刘小飞，靳引利
（长安大学 电子与控制工程学院，陕西 西安 710064）

**摘 要**：针对考虑电量限制的电动车辆在同时取送货场景下的路径规划问题，提出一种改进的文化基因算法．该算法采用基于极坐标的初始化方法生成初始解，引入一种最优分割方法优化解的质量，并设计多种特定的局部搜索算子提升搜索效率．同时，通过扰动机制避免陷入局部最优．将所提局部算子集成到其他算法中，在小、中、大规模数据集上进行对比实验，并结合Wilcoxon符号秩检验对不同算法进行性能分析．结果表明，该算法在各项评价指标上均优于对比算法，且提出的局部搜索算子加入后算法性能显著提升，验证了其在求解优质解方面的有效性．

**关 键 词**：车辆路径规划；同时取送货；电动汽车；文化基因算法；最优分割方法
**中图分类号**：TP 18 **文献标志码**：A **文章编号**：1005-3026(2026)04-0016-08

# Algorithm for Electric Vehicle Routing in Simultaneous Pickup and Delivery Scenarios

*LI Xiao-hui*, *SUN Wei-tong*, *LIU Xiao-fei*, *JIN Yin-li*
(School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China. Corresponding author: JIN Yin-li, E-mail: 915785835@qq.com)

**Abstract**: An improved memetic algorithm was proposed to address the electric vehicle routing problem with simultaneous pickup and delivery under battery capacity constraints. The algorithm employed a polar coordinate-based initialization method to generate initial solutions and introduced an optimal splitting method to enhance solution quality. Multiple specialized local search operators were designed to improve search efficiency, and a perturbation mechanism was incorporated to escape local optima. The proposed local search operators were integrated into other algorithms, and comparative experiments were conducted on small-, medium-, and large-scale datasets. Performance evaluations were performed using the Wilcoxon signed-rank test. The results demonstrate that the proposed algorithm outperforms comparative algorithms across all evaluation metrics. Additionally, the performance of other algorithms significantly improves after incorporating the proposed local search operators, confirming their effectiveness in obtaining high-quality solutions.

**Key words**: vehicle routing; simultaneous pickup and delivery; electric vehicle; memetic algorithm; optimal splitting method

自Dantzig等[1]首次提出经典的车辆路径规划问题（vehicle routing problem，VRP）以来，学术界对这一问题及其各种变体开展了广泛的研究．VRP的基本目标是在满足约束条件的前提下，为车辆规划客户服务的最优路径，问题的复杂性来自处理与车辆容量、待收集产品的性质、时间窗口和需求条件有关的若干限制因素[2-3]．

近年来，取送货的VRP问题由于在实际应用中，特别是逆向物流中的重要性，得到了越来越多研究者和从业者的关注．提出了同时取送货的VRPSPD（vehicle routing problem with simultaneous pickup and delivery），以解决图书馆之间的图书运输问

**收稿日期**：2025-02-26
**基金项目**：国家自然科学基金资助项目（72471188）．
**作者简介**：李晓辉（1982—），男，陕西西安人，长安大学副教授．
**通信作者**：靳引利，E-mail：915785835@qq.com.

题[4].Cavaliere 等[5]为 VRPSPD 和混合取送货的 VRPMPD（vehicle routing problem with mixed pickup and delivery）设计了一种针对性的算法，该算法可解决这类问题的大规模实例.同时，电动车辆路径规划问题（electric vehicle routing problem，EVRP）作为VRP的一类变体，涉及到为电动汽车设计最优路线，同时考虑电池约束和充电操作[6].Duman等[7]提出了带时间窗的EVRP模型，并设计了精确算法与基于列生成的启发式方法，以提高求解效率.Goeke[8]研究了电动汽车的取货与送货问题，提出了一种粒度禁忌搜索方法，并针对不同规模的案例进行了数值实验.Qian等[9]研究了具有时间窗和电池交换站的电动汽车路径规划问题，为该问题提出了一种交替方向乘法器方法和可变邻域搜索算法.

在VRP及其变种的求解过程中，研究者们提出了多种启发式算法与精确算法[10-11].精确算法从理论上保证了在有限时间内获得最优解，但在实际应用中，由于计算量庞大，往往存在计算耗时过长的问题.相比之下，启发式算法被证明在解决VRP问题时，能够以更高的效率和良好的性能获得近似最优解.例如，粒子群算法[12]、遗传算法[13]、蚁群算法[14]等在实际应用中表现出色.本文提出了一种改进的文化基因算法（improved memetic algorithm，IMA）来解决电动汽车的同时取送货路径规划问题（electric vehicle routing problem with simultaneous pickup and delivery，EVRPSPD）.该算法对传统文化基因算法进行了优化，并通过与几种经典启发式算法的对比实验验证了其可靠性与有效性.

## 1　问题描述

某一地区的配送中心为一定数量的客户提供服务，每个客户有不同的取送货需求.一定数量的电动汽车从配送中心出发，访问客户节点，满足客户相应的取送货需求，然后返回配送中心.此外，在整个行程中，当电动汽车能量不足时，需要访问充电服务站CS进行充电，电能消耗主要取决于行驶距离.将这一问题定义为同时取货和送货的电动车辆路径规划问题.为了更好地描述这一问题，以一个实例作说明，见图1.

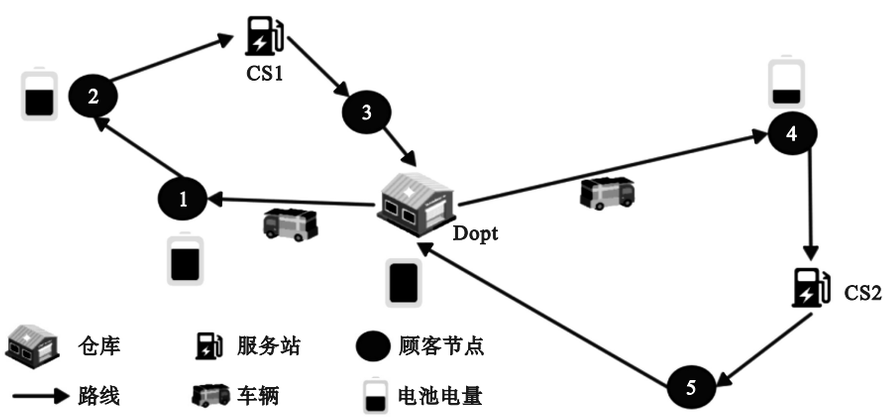



图1　问题描述

Fig. 1　Problem description

在这个例子中，有1个配送中心（仓库）、5个客户和2个服务站CS.现有若干辆电动汽车从仓库出发，需要在满足自身容量和能量限制的条件下，完成所有客户节点的取送货需求，并返回仓库；中间能量不足时，允许寻找最近的服务站充电.该问题的目标是使所有车辆的总行驶距离最短.在数学模型中，为简化问题，作出以下假设：

1）所有车辆从仓库出发，配送完成后必须返回仓库；

2）所有的顾客节点必须被车辆访问到，并且每个顾客节点只能被访问一次；

3）车辆在配送过程中，可以通过访问服务站补充能量以增加续航里程.车辆访问服务站的次数可以是一次、多次，也可以不访问.本文不考虑补充能量的形式，即车辆访问服务站后为充满状态；

4）在该问题中不限制车辆的使用数量，即假定仓库车辆的个数为无穷大；

5）在本问题中，车辆消耗能量的速率为常数，能量的消耗量只与行驶路程有关；

6）每个顾客节点同时有取货和送货的需求，车辆在出发时，需要带上计划行驶路线上所有顾客节点的配送量的货物，即取货货物和送货货物相互独立；

7) 要保证车辆在行驶过程中,其载货量始终不超过其容量限制;

8) 车辆在配送过程中,其能量不可降至0以下.

## 2 算法介绍

针对EVRPSPD问题,本文提出了一种改进的文化基因算法(IMA)来解决.需要先读入顾客节点信息、车辆参数等数据作为输入量,然后进入算法流程.

该算法首先利用3种不同的生成个体的方式初始化种群,然后通过将种群中选择出来的特定的2个个体进行交叉进化的方式,增加种群的多样性.之后,基于精英保留策略,只有当个体更新后距离路径总距离小于原个体时,才对其进行更新,否则还原为原个体.当种群若干代最优解没有变化时,即判定达到了扰动的条件.此时种群可能陷入了局部最优,通过扰动的方式可以使其解空间发生变化,帮助跳出局部最优.若加入扰动后结果仍未有改进且满足迭代终止条件,则判定为收敛,终止算法.流程图见图2,伪代码见算法1.

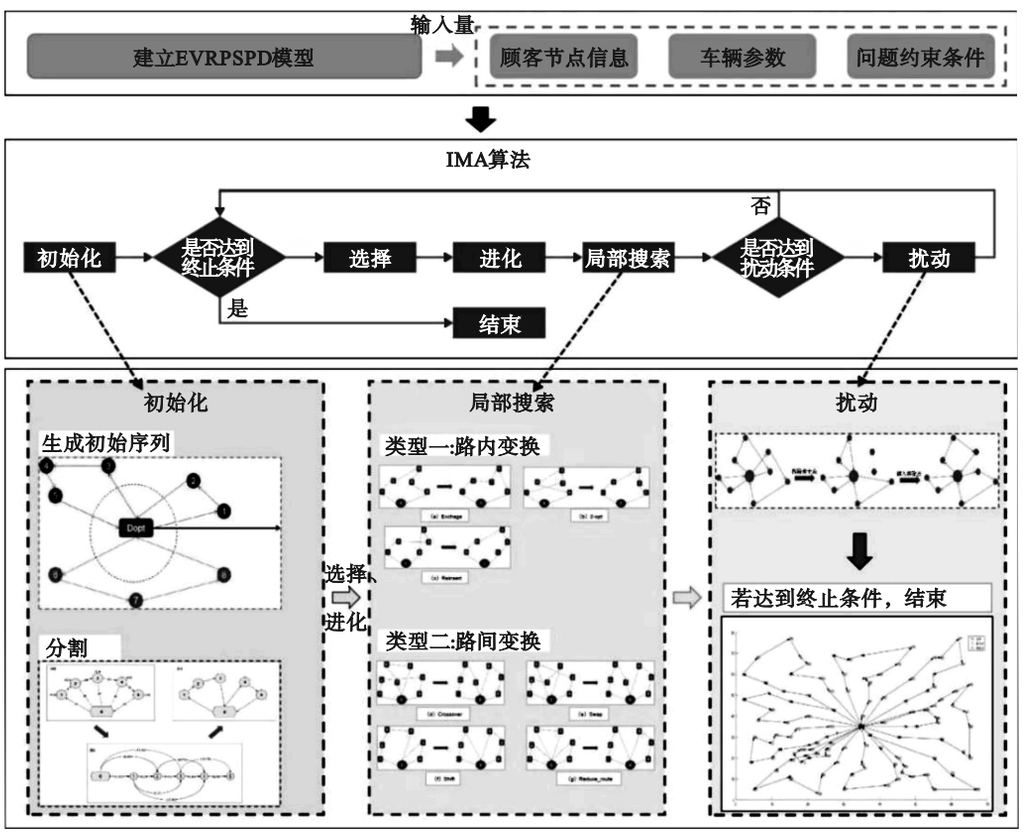



图2 算法整体框架

Fig. 2 Framework of algorithm

### 2.1 编 码

每个解有两种形式,见图3.第1种编码方式命名为type1,不包含仓库节点和充电服务站CS,仅包含各个顾客节点的相对序列信息,例如3-1-2-4-5.第2种编码方式命名为type2,包含仓库节点和充电服务站CS,为实际路过节点的路线,例如0-2-6-0-3-1-4-7-0-5-8-0.两种形式可以通过分割(split)和更新(update)相互转换.

type1和type2之间的转换通过对type1的分割实现.首先,对type1按照容量进行分割,按照容量的分割方式需要保证整个配送过程中车辆的容量不超过其最大容量;然后,将分割完的序列按能量分割,将服务站加入序列中,选择的服务站为到被分割的两点之间距离最短的服务站,同时要保证分割的上一个节点的剩余能量能够到达服务站.分割策略主要有:基于贪心策略的分割方式、基于Prins[15]的最优分割方式.

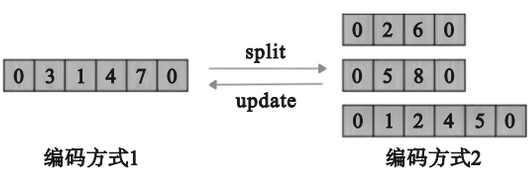



图3 编码方式

Fig. 3 Encoding method

贪心策略是基于约束条件进行分割,当达到约束条件阈值时,对序列进行分割.

基于Prins的最优分割方式见图4,其在满足约束条件的基础上,寻求最优的分割方式.图4a为一条type1的节点分布情况,节点外的数字分别代表该节点的送货需求和取货需求,每条线段上的数字则是两个节点之间的距离.可以将每条分割路线转化为有向图(图4b),图中每条路线都代表了一条分割出来的路线,路线上的数字分别代表该路线的最大负荷和路线成本,不存在的路线则代表该分割路线超出约束条件.通过迪杰斯特拉算法,就能找出覆盖所有节点的成本最小的路线,从而确定总成本最小的最优分割方式.

算法1　改进的文化基因算法

输入:问题所需要的参数和城市坐标需求信息

输出:近似最优解$X$

**Begin**

//初始化种群

**for** $i$=1:num

依据初始化方式生成分割前的初始解$X_{type1}$,用最优分割方式对初始解进行分割得到解$X_{type2}$,计算种群中所有解的适应度$fitness_i$,记录种群最优解$X_{best}$

**end**

**While**(不满足终止条件)　**do**

//对种群个体进行选择、进化、局部搜索操作

**for** $i$=1:num

计算种群中所有解的适应度$fitness_i$,并依据适应度计算每个个体的选择概率

依据选择概率选择待交叉的种群,将每个个体的$X_{type1}$与其进行交叉操作

对每个个体$X_{type2}$进行局部搜索,当搜索到的新个体优于原个体时,用其更新$X_{type2}$

**end**

更新种群最优解$X_{best}$;

//扰动阶段

**if**($X_{best}$若干次没有更新)**then**对种群进行扰动,启动重启机制

**end**

返回当前种群最优解$X_{best}$

**end**

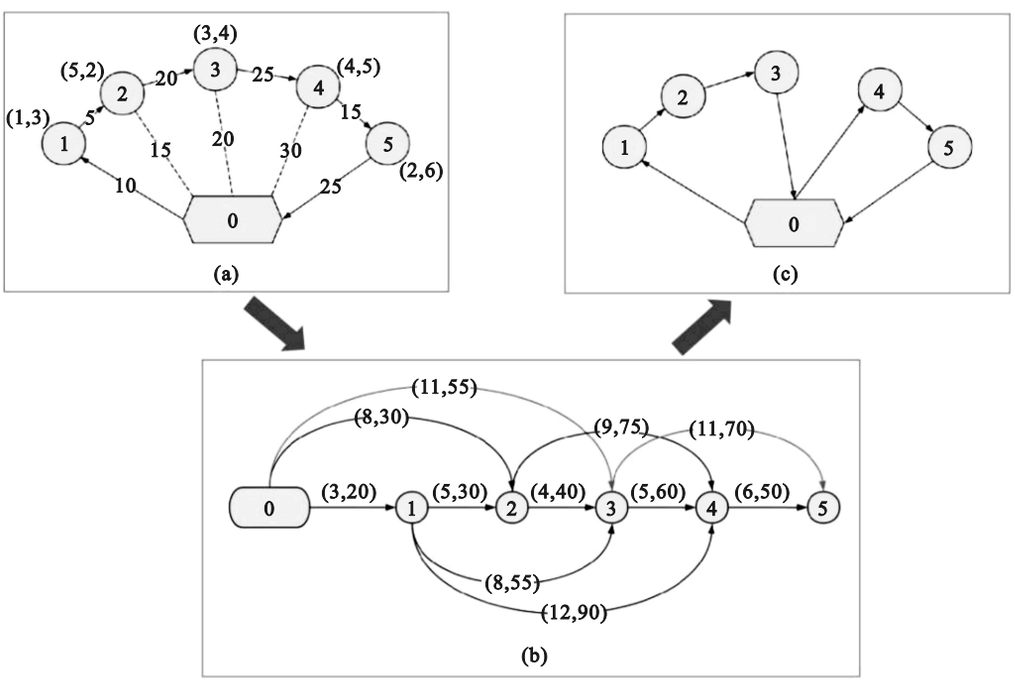



图4　最优分割方式

Fig. 4　Optimal splitting method

(a)—节点的分布;(b)—有向图;(c)—分割完的路线.

最优分割算法流程及伪代码如算法2所示.该算法的第1部分是为type1路线$X$的每个节点$j$=1,2,…,$|N_0|$计算2个标签(即$V_j$和$P_j$),而不是明确生成对应的type2.$V_j$代表type2中从节点0到节点$j$的最短路径的成本,$P_j$则是节点$j$在这条路径上的前驱节点.循环在枚举所有可行的子序列$X(i)$,…,$X(j)$之后更新$V_j$和$P_j$.需要注意的是,对于给定的节点$i$,$j$会一直增加,直到车辆的最大负

载(max_load)超过车辆容量($Q$).算法的第2部分使用前驱节点$P$提取分割后的type2方案.

算法2 最优分割

输入:type1类型的路线$X$

输出:type2类型的路线$X$

**Begin**

//计算从仓库点到各个节点的最短路线的相关变量($V_j$和$P_j$),路线每个节点$j$//

$V_0 \leftarrow 0; V_j \leftarrow \infty \forall j \in \{1,2,\cdots,|N_0|\}$

**for** $i=1,2,\cdots,|N_0|$ **do begin**

max_load←0;pload←0;sload←0;$j \leftarrow i$

**repeat**

pload←max_load+$d_{X(j)}$;

sload←sload+$p_{X(j)}$;

max_load←max{pload,sload};

**if** $i=j$ **then** cost←$c_{(0,X(i))}+c_{(X(i),0)}$

**else** cost←cost−$c_{(X(i),0)}+c_{(X(i),X(j))}+c_{(X(j),0)}$

**if** max_load≤$Q$ **then begin**

**if** ($V_{i-1}$+cost<$V_j$) **then begin**

$V_j \leftarrow V_{i-1}$+cost

$P_j \leftarrow i-1$

**end**

$j \leftarrow j+1$

**end**

**Until** $j>|N_0|$ or max_load>$Q$

**end**

//使用前驱节点数组$P$列出type2路线

**for** $i=1,2,\cdots,|N_0|$ **do** route($i$)←∅

$r \leftarrow 0; j \leftarrow |N_0|$

**repeat**

$r \leftarrow r+1; i \leftarrow P_j$

**for** $k=i+1$ **to** $j$ **do** route($r$)←route($r$)∪{$X(i)$}

**end for**

$R \leftarrow R \cup$ route($r$)

$j \leftarrow i$

**until** $i=0$

返回车辆路径集$R$

**end**

## 2.2 初始化

为保证生成的初始解具有优越性的同时又具有一定的多样性.本文设计了3种生成初始解的方案:

1) 随机生成初始解.随机生成一段序列,将其顺序打乱作为初始解.

2) 基于最近邻域的选择法.将随机点确定为初始点,然后从剩余未访问的顾客节点中选择一个距离之前访问点最近的点作为下一个被访问的节点,直至所有节点被选择完.

3) 基于经典扫描算法的极坐标体系,以仓库点作为原点,先算出各个节点的极坐标,将随机点作为初始点,每访问一个节点后,下一个节点选择与其极坐标最相近的节点,直至所有节点被选择完,如图5所示.

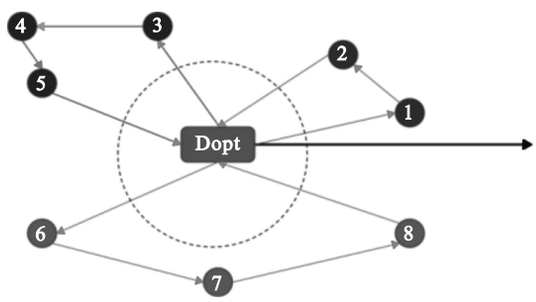



图5 极坐标初始化

Fig. 5 Polar coordinate-based initialization

基于以上方式得到了顾客节点的相对序列,即type1,然后使用上文提到的分割方式得到初始解,从而构建初始种群.

## 2.3 计算适应度

本文的优化目标为所有配送路线的总距离最小,因此使用配送路线距离的倒数作为每个个体的适应度值,并将适应度作为选择算子的参考指标.

## 2.4 选 择

在选择个体时,应保证适应度大的解有较大的概率被选择,同时也应使适应度较小的个体有被选择的可能.基于以上原则,本文选取了轮盘赌策略进行个体的选择.每个个体被选择的概率基于其适应度,计算出其适应度占种群所有个体适应度之和的比例,该比例就是每个个体被选择的概率,以此选出后续执行进化操作的个体.

## 2.5 进 化

进化操作采用交叉的方式,具体为顺序交叉,操作对象为type1类型.从父代中随机选择起止位置,保留该部分父代的基因序列,并从母代中选取父代中没有的基因将该序列补充完整,并保留补充序列的相对顺序.另一个子代也以同样的方式生成,具体操作流程见图6.

## 2.6 局部搜索

局部搜索针对路内、路间有不同的策略,操作对象均为type2类型,操作示意图见图7.局部

搜索得到的解有可行解和不可行解,当遇到不可行解时,对其进行修复:将其恢复为type1类型,再对其进行重新分割,得到新的type2解.

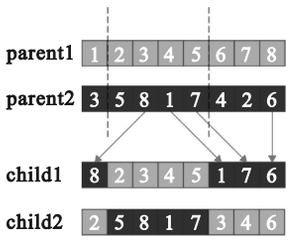



图6　顺序交叉

Fig. 6　Order crossover

2.6.1　路内变换

Exchange:选择一条子路径$r1$,从$r1$中选择两个节点$a$,$b$,将其交换位置.

2-opt:选择一条子路径$r1$,从$r1$中选择一段目标序列$s1$,对其进行逆序操作.

Reinsert:选择一条子路径$r1$,从$r1$中选择一个节点$a$,将$a$从$r1$中移除,然后将其重新插入到$r1$中的最优位置.最优位置即该节点插入的所有位置中,插入后的总距离最小的位置.

Reverse:该操作将一条子路径整体逆序,这样并不改变该条路径的距离,但由于存在容量限制或者车辆能量不足的情况,该操作可能会降低某条路径的最大容量,也可能使某些因能量问题导致的非法路径变为合法路径.

2.6.2　路间变换

Crossover:该操作选择解中的任意两条子路径$r1$和$r2$,将它们分别拆分为两部分,之后将$r1$和$r2$的第1部分保留,第2部分相互替换.

Swap:首先选择两条子路径$r1$和$r2$,然后从$r1$中选择该配送路线中的一个节点$a$,从$r2$中选择该配送路线中的一个节点$b$,分别将其插入到另一条子路径中的最优位置,即将$a$插入到$r2$中的最优位置,将$b$插入到$r1$中的最优位置.

Shift:首先选择一条子路径$r1$,然后移除$r1$中的一个节点,将其重新插入到其他子路径而不是该节点原来所在的子路径.

Reduce_route:该操作用来缩减最终配送方案中的路径数目,从而优化路径的总距离.首先从当前解中选择一条子路径$r1$作为待拆解路径,然后尝试将$r1$中的节点逐个插入到其他子路径中.

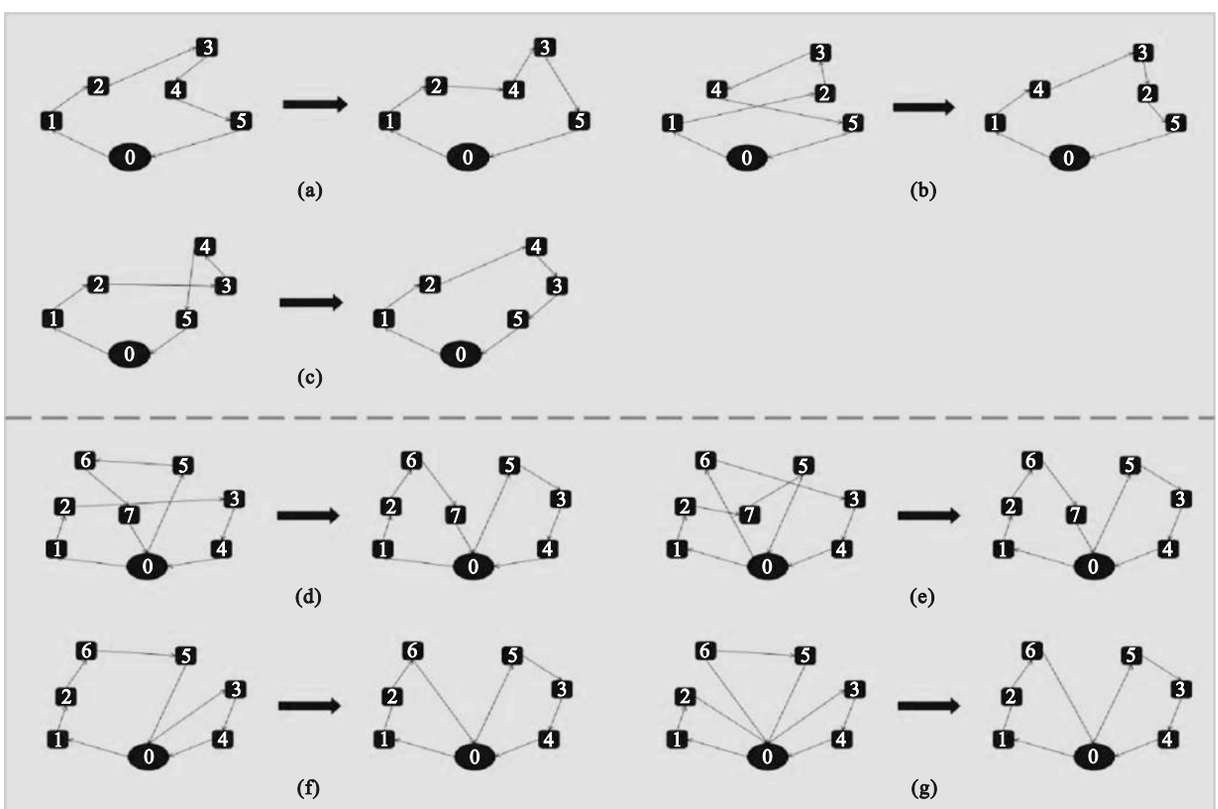



图7　局部搜索算子

Fig. 7　Local search operator

路内变换:(a)—Exchage;(b)—2-opt;(c)—Reinsert;

路间变换:(d)—Crossover;(e)—Swap;(f)—Shift;(g)—Reduce_route.

### 2.7 扰　　动

本文引入的扰动并非每次迭代都会执行,是为了避免陷入局部最优的情况.对迭代到一定次数后仍未出现新的最优解的种群,启动重启机制,即通过扰动使部分重复的解和较差的解进行重启,只保留前20%的优秀解不变.扰动使用了一对拆除-插入操作算子(dr_random),将需要重启的解中的若干个顾客节点移除,移除的数量等于顾客节点总数除以车辆个数,将这些节点插入到最佳的位置,从而产生新的优秀解.这样产生的解既保证了与原解有较大的差异,同时又保证了优越性,便于种群的继续搜索.

## 3 对比实验

本文提出的算法使用C++实现,编译器采用Visual Studio 2022,操作系统为Windows 10,CPU为AMD Ryzen7 3750H 2.30 GHz,内存16 GB.本次实验在3种规模的数据集上进行,并使用了其他3种算法进行对比,并且为了证明本文提出的局部搜索算子的有效性,对其中一种算法使用局部搜索进行了改进,将其改进前后进行实验对比.然后通过最优解、平均值以及标准差等指标评估算法的性能.

1)基准数据集.本文数据集基于EVRP[16]的数据集生成,在原数据集的基础上,修改了其每个点的需求,使其拥有取货和送货的需求,从而满足本文的数据集要求,见表1.

2)对比算法.为了证明算法的稳定性,本文在小、中、大三种规模的数据集上进行了测试,并且与模拟退火算法(simulated annealing,SA),粒子群优化算法(particle swarm optimization,PSO),蚁群优化算法(ant colony optimization,ACO)进行了对比,由于传统蚁群算法在该问题上效果欠佳,本文用所提出的邻域搜索算子对ACO进行了改进,即ACO_improved,将其与传统ACO进行对比,以验证本文提出的邻域搜索算子的有效性.除了基于单解的SA,基于种群的算法包括IMA、PSO、ACO,其种群中个体数均为100.统计量包括各个算法的最优解、运行的平均值,为了验证本文算法的稳定性,还对IMA求了标准差,以确定其稳定性.为了评估算法性能,采用了Gap指标,其计算方法为IMA的最优解和对比算法的最优解的差距,Gap为负证明IMA的最优解优于其他算法,为正则说明其他算法存在优于IMA的情况,实验结果见表2.

**表1 基准数据集**

Table 1 Benchmark datasets

| 名称 | 顾客节点 | 充电站 | 容量 | 电量 | 名称 | 顾客节点 | 充电站 | 容量 | 电量 |
|---|---|---|---|---|---|---|---|---|---|
| P1 | 21 | 7 | 6 000 | 99 | P9 | 71 | 8 | 30 000 | 53 |
| P2 | 22 | 7 | 4 500 | 162 | P10 | 134 | 5 | 2 210 | 307 |
| P3 | 29 | 5 | 4 500 | 138 | P11 | 100 | 9 | 200 | 118 |
| P4 | 32 | 4 | 8 000 | 238 | P12 | 120 | 5 | 200 | 199 |
| P5 | 50 | 9 | 160 | 88 | P13 | 150 | 12 | 200 | 100 |
| P6 | 75 | 13 | 220 | 87 | P14 | 199 | 12 | 200 | 100 |
| P7 | 100 | 11 | 200 | 100 | P15 | 142 | 4 | 1 190 | 2 762 |
| P8 | 44 | 4 | 2 010 | 260 | P16 | 213 | 7 | 944 | 1 204 |

**表2 实验结果**

Table 2 Experimental results

| ID | SA | | PSO | | ACO | | ACO_I | | IMA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Best | Avg | Best | Avg | Best | Avg | Best | Avg | Best | Avg | SD | Gap/% |
| P1 | 404.0 | 428.4 | 392.7 | 401.3 | 407.6 | 410.5 | 400.3 | 400.3 | **392.7** | 397.8 | 2.56 | 0.00 |
| P2 | 716.7 | 762.2 | 712.4 | 716.7 | 710.5 | 713.4 | 709.7 | 712.9 | **709.7** | 710.7 | 0.77 | 0.00 |
| P3 | 555.8 | 556.4 | 551.8 | 553.5 | 563.4 | 568.9 | 552.9 | 555.8 | **551.7** | 552.1 | 0.74 | −0.02 |
| P4 | 1 356.2 | 1 573.3 | **1 206.8** | 1 226.2 | 1 293.4 | 1 319.2 | 1 243.4 | 1 268.5 | 1 216.0 | 1 230.4 | 9.89 | 0.76 |
| P5 | 553.2 | 568.3 | **528.5** | 542.2 | 601.0 | 617.6 | 586.7 | 605.7 | 535.7 | 546.9 | 8.42 | 1.35 |
| P6 | 762.6 | 788.1 | 767.8 | 780.2 | 828.3 | 847.6 | 806.1 | 827.4 | **756.6** | 767.1 | 6.12 | −0.79 |
| P7 | 987.2 | 1 005.4 | 986.6 | 998.5 | 1 069.2 | 1 108.3 | 1 021.3 | 1 048.9 | **972.0** | 982.6 | 5.44 | −1.50 |
| P8 | 847.4 | 939.9 | 846.9 | 861.4 | 924.2 | 951.7 | 905.4 | 930.3 | **843.4** | 851.1 | 7.50 | −0.42 |
| P9 | 292.9 | 309.7 | **281.3** | 298.2 | 320.0 | 326.0 | 299.7 | 310.2 | 283.7 | 289.8 | 4.44 | 0.86 |
| P10 | 1 386.3 | 1 386.3 | 1 225.3 | 1 279.6 | 1 364.3 | 1 413.5 | 1 301.1 | 1 340.6 | **1 216.8** | 1 263.9 | 19.23 | −0.69 |
| P11 | 1 668.2 | 1 718.6 | 1 648.2 | 1 674.0 | 1 691.8 | 1 722.4 | 1 662.1 | 1 680.4 | **1 602.3** | 1 651.4 | 19.15 | −2.86 |
| P12 | 1 735.8 | 1 823.6 | 1 623.6 | 1 702.6 | 1 662.8 | 1 680.4 | **1 616.5** | 1 639.1 | 1 630.9 | 1 698.4 | 28.31 | 0.89 |
| P13 | 1 590.2 | 1 624.2 | 1 569.0 | 1 588.3 | 1 638.2 | 1 657.2 | 1 593.6 | 1 612.0 | **1 546.6** | 1 569.8 | 11.34 | −1.45 |
| P14 | 1 923.5 | 1 961.3 | 1 886.4 | 1 921.3 | 1 956.3 | 1 991.5 | 1 924.5 | 1 946.8 | **1 876.3** | 1 897.7 | 11.06 | −0.54 |
| P15 | 19 772 | 19 772 | **18 791** | 19 299 | 19 998 | 20 632 | 19 399 | 19 729 | 18 926 | 19 108 | 116.52 | 0.71 |
| P16 | 15 408 | 15 408 | 15 169 | 15 346 | 15 363 | 15 640 | 15 061 | 15 357 | **15 010** | 15 189 | 98.89 | −0.34 |

实验结果表明,在所有实验中IMA均优于SA、ACO.本文提出的IMA算法相比于SA,其最优值和平均值分别提升了3.64%和5.75%;相比于PSO,IMA在大部分实例上的最优值超过PSO,有4个实例没有超过,但其最优值和平均值分别提升了0.38%和0.95%;相比于ACO,最优值和平均值分别提升了5.99%和6.31%;相比于ACO_I,IMA仅有一个实例未超过,但其最优值和平均值分别提升了3.35%和3.51%.

本文还使用了Wilcoxon符号秩检验来验证提出算法的性能,结果如表3所示.表中,*w*/*t*/*l*代表在16个数据集中两种对比算法的赢/平/输的个数.可以看出,IMA在所有情况下都能获得比*R*−值更高的*R*+值,意味着IMA在大多数情况下比其他算法保持更好的收敛性和近似值分布.同时,ACO_I在所有的测试中,其结果都优于ACO,证明了本文提出的邻域搜索算子的可行性.

表3　算法性能比较统计

Table 3　Comparative statistics of algorithm performance

| 算法 | *w*/*t*/*l* | *R*+ | *R*− | *p*−value | *α* | |
|---|---|---|---|---|---|---|
| | | | | | 0.05 | 0.15 |
| IMA vs SA | 16/0/0 | 317 | 3 | 0.0067 | TRUE | TRUE |
| IMA vs PSO | 10/1/5 | 243 | 70 | 0.0371 | TRUE | TRUE |
| IMA vs ACO | 16/0/0 | 294 | 15 | 0.0232 | TRUE | TRUE |
| IMA vs ACO_I | 14/1/1 | 281 | 29 | 0.0407 | TRUE | TRUE |

## 4　结　　论

1)针对EVRPSPD的求解,提出了一种IMA算法,引入了扰动机制以跳出局部最优.使用基于极坐标的初始化方式和最优分割方法,提升了初始解的质量.设计了几种特定的局部搜索算子,提高了对新解的搜索效率和搜索质量.

2)实验结果和Wilcoxon符号秩检验分析表明,与ACO等传统算法相比,本算法具有较好的寻优能力.局部搜索算子加入算法前后的对比结果表明,设计的局部搜索算子在解的搜索方面具有较高的效率和质量,证明该算法在解决物流配送等实际问题方面可以提供参考.

**参考文献:**


[1] Dantzig G B, Ramser J H. The truck dispatching problem [J]. *Management Science*, 1959, 6(1): 80−91.

[2] Gutiérrez A, Rocha L B. VRP variants applicable to collecting donations and similar problems: a taxonomic review[J]. *Computers & Industrial Engineering*, 2022, 164(3): 107887.

[3] Roudbari E S, Ghomi S M T F, Sajadieh M S. Reverse logistics network design for product reuse, remanufacturing, recycling and refurbishing under uncertainty[J]. *Journal of Manufacturing Systems*, 2021, 60(14): 473−486.

[4] Hokey M. The multiple vehicle routing problem with simultaneous delivery and pick-up points [J]. *Transportation Research Part A General*, 1989, 23(5): 377−386.

[5] Cavaliere F, Accorsi L, Laganà D, et al. An efficient heuristic for very large-scale vehicle routing problems with simultaneous pickup and delivery [J]. *Transportation Research Part E*, 2024, 186(4): 103550.

[6] Yang B H, Ren T, Yu H J, et al. An evolutionary algorithm driving by dimensionality reduction operator and knowledge model for the electric vehicle routing problem with flexible charging strategy [J]. *Swarm and Evolutionary Computation*, 2025, 92: 101814.

[7] Duman E N, Taş D, Çatay B. A bidirectional branch-and-price algorithm with pulse procedure for the electric vehicle routing problem with flexible deliveries[J]. *Transportation Research Part C: Emerging Technologies*, 2024, 165: 104699.

[8] Goeke D. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles [J]. *European Journal of Operational Research*, 2019, 278(3): 821−836.

[9] Qian B, Feng F L, Yu N K, et al. An alternating direction multiplier method with variable neighborhood search for electric vehicle routing problem with time windows and battery swapping stations [J]. *Applied Soft Computing*, 2024, 166: 112141.

[10] Xu S, Ou X Y, Govindan K, et al. An adaptive genetic hyper-heuristic algorithm for a two-echelon vehicle routing problem with dual-customer satisfaction in community group-buying [J]. *Transportation Research Part E*, 2025, 194: 103874−103874.

[11] Wang S H, Sun W, Huang M, et al. An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time windows [J]. *Computers & Industrial Engineering*, 2024, 191(15): 110122.

[12] Wu Q C, Xia X W, Song H J, et al. A neighborhood comprehensive learning particle swarm optimization for the vehicle routing problem with time windows[J]. *Swarm and Evolutionary Computation*, 2024, 84: 101425.

[13] Wang C, Ma B, Sun J Y. A co-evolutionary genetic algorithm with knowledge transfer for multi-objective capacitated vehicle routing problems [J]. *Applied Soft Computing*, 2023, 148: 110913.

[14] Zhu S L, Wang C. An interaction-enhanced co-evolutionary algorithm for electric vehicle routing optimization [J]. *Applied Soft Computing*, 2024, 165: 112113.

[15] Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem [J]. *Computers & Operations Research*, 2004, 31(12): 1985−2002.

[16] Mavrovouniotis M, Menelaou C, Timotheou S, et al. A benchmark test suite for the electric capacitated vehicle routing problem[C]// 2020 IEEE Congress on Evolutionary Computation. Glascow, 2020: 1−8.