

改进野马算法求解低碳开放式送取货选址路径问题

虎翼飞, 张惠珍*, 陈曦
(上海理工大学 管理学院, 上海 200093)

摘要: **目的** 针对当前物流背景下普遍出现的送货公司外包、退换货频繁等问题, 结合现有的碳排放政策, 提出低碳背景下开放式同时送取货选址-路径模型 (Low-Carbon Open Location-routing Problem with Simultaneous Pickup and Delivery Problem, LOLRSPD), 并通过改进野马算法进行求解。**方法** 首先设计一种新的解码方式, 使得原离散问题可以采用连续算法求解。之后, 运用哈尔顿序列生成初始解, 改进非线性进化概率因子, 使用模拟二进制交叉, 增加变异操作, 以及精英保留、设置连续失败重新初始化等步骤, 改进野马算法。最后, 通过6组不同大小的算例将改进野马算法与原始野马算法、模拟退火算法、粒子群算法、遗传算法进行对比。**结果** 针对中大型算例, 改进野马算法远超原始野马算法。针对小型算例, 在确保准确率的同时, 改进野马算法对比各经典算法也在速度上具有优势。**结论** 提出的LOLRSPD模型具备合理性, 改进的野马算法针对选址路径问题具有较好的搜索能力。

关键词: 选址路径问题; 开放式问题; 同时送取货; 改进野马算法; 元启发式算法

中图分类号: TP302; TB48 **文献标志码:** A **文章编号:** 1001-3563(2024)01-0229-10

DOI: 10.19554/j.cnki.1001-3563.2024.01.027

Improved Wild Horse Optimizer for Solving Low-carbon Open Location-routing Problem with Simultaneous Pickup and Delivery Problem

HU Yifei, ZHANG Huizhen*, CHEN Xi

(School of Management, University of Shanghai for Science & Technology, Shanghai 200093, China)

ABSTRACT: The work aims to propose an Open Location-Routing Problem with Simultaneous Pickup and Delivery (LOLRSPD) model under the low-carbon background to solve the common problems of outsourcing, frequent return and exchange of goods in delivery companies and solve the problem through the Wild Horse Optimizer. Firstly, a new decoding mode was designed so that the discrete problems could be solved by continuous algorithms. Then, the initial solutions were generated through the Halton sequence to improve the nonlinear evolution probability factor TDR; The simulated binary crossover was used. Polynomial mutation operators were added. Elite preservation, and setting of consecutive failed reinitialization steps were completed. Finally, the effectiveness of the improved algorithm was verified through the comparison results between Improved Wild Horse Optimizer (IWHO), Wild Horse Optimizer (WHO), Simulated Annealing (SA), Particle Swarm Optimization (PSO), and genetic algorithm (GA). The experimental results showed that IWHO in this paper had better optimization ability than normal WHO in terms of large and medium-sized examples, and had a good advantage in the processing of small examples while ensuring accuracy. The proposed LOLRSPD model is reasonable, and the Improved Wild Horse Optimizer has better searching ability for LRP problems.

KEY WORDS: location-routing problem; open location-routing problem; simultaneous pickup and delivery; improved wild horse optimizer; heuristic algorithm

收稿日期: 2023-05-17

基金项目: 国家自然科学基金 (72101149); 教育部人文社会科学基金 (21YJC630087)

*通信作者

选址路径问题 (Location-routing Problem, LRP) 是一类经典的组合优化问题, 需要同时将设施点的选址及车辆路线的选择纳入决策。中国作为人口大国, 物流行业规模庞大, 且发展迅速。相对而言, 我国在物流研究方面仍较落后, 大多企业只能依靠经验来制定物流标准, 使得物流服务缺乏创新。随着互联网的普及, 更多中老年群体开始使用手机购物, 使得快递名目更繁杂, 且退换货的比例加大。将开放式与同时送取货组合对增强企业竞争优势、减少社会不必要浪费具有显著意义。

Cooper^[1]首次将设施选址问题与车辆路径问题结合起来, 提出了 LRP。将标准 LRP 与实际情况结合后, 可以形成多种扩展 LRP, 如有容量限制的 LRP (Capacitated LRP, CLRP)、带车容量限制的选址路径问题 (Capacitated Location-Routing Problem, CLRP)、开放式选址路径问题 (Open Location-routing Problem, OLRP)、同时送取货选址路径问题 (Location-Routing Problem with Simultaneous Pickup and Delivery, LRPSPD) 等。已有大量学者对 LRP 及其扩展问题进行了研究。Karaoglan 等^[2-4]根据实际运输网络中同时送取货的需要, 提出了同时送取货选址路径问题 (Location Routing Problem with Simultaneous Pickup and Delivery, LRPSPD), 并分别使用混合遗传算法、分支定界法、启发式算法进行求解。Wang^[5]提出了具有多种服务模式和模糊时间窗的 LRPSPD, 使用禁忌搜索算法进行求解验证, 作为单起点式算法, 其求解速度相对较慢, 未发挥启发式算法的优势。刘亚平等^[6]在同时送取货的基础上, 考虑了客户时间窗, 建立了 LRPSPD_{TDW} 模型, 并用改进的烟花算法求解。通过与 B&C (Branch and Cut algorithm) 获得的最优解进行比较, 取得了不错的成果, 但在大规模算例中缺少与其他启发式算法的对比实验。刘东等^[7]在同时送取货的基础上, 考虑了开放式, 即车辆配送后不返回仓库, 提出了 OLRPSPD (Open Location-routing Problem with Simultaneous Pickup and Delivery Problem) 模型, 进一步增大了现实意义。还改进了蘑菇算法, 相较于模拟退火算法、蚁群算法、混合免疫算法等经典启发式算法, 此改进方法在中型算例中具有明显的优势。

随着多项节能减排政策的实施和中国碳排放权交易市场的建立, 物流活动中的碳排放成为一个亟待解决的关键问题。为此, 国内外学者们通过在 LRP 中加入低碳要素, 通过构建低碳模型来减少碳排放量。Barth 等^[8]考虑了车辆的速度和装载量对车辆油耗的关系, 建立了货车燃油消耗模型。Zhang 等^[9]考虑了行驶距离、汽车行驶速度、车辆装载量等,

假设燃油消耗随着车辆的载质量呈等比例增加, 提出了燃油消耗模型。由于在 LRP 中很难具体描述车辆的速度和加速度, 故 Xiao 等^[10]假设车辆匀速行驶, 只考虑载质量和行驶距离对车辆燃油排放的影响。文中在 OLRPSPD 模型的基础上加入油耗计算公式, 建立 LOLRPSPD (Low-Carbon Open Location-routing Problem with Simultaneous Pickup and Delivery Problem) 模型。

野马优化算法 (Wild Horse Optimizer, WHO) 是 Naruei 等^[11]在 2021 年提出的一种智能优化算法, 该算法的灵感来自野马的放牧、交配和群体领导行为。通过模拟野马放牧过程中马驹在领头马周围游荡和不同马驹群体间交配的特点, 将领头马和马驹群体位置的变化过程作为算法的优化过程。WHO 具有参数少、寻优能力强、时间复杂度低等优点。该算法仍然存在种群初始化阶段种群多样性差、全局搜索能力弱、后期难以脱离局部最优等问题。针对基本 WHO 的不足, 文中提出一种混合多策略改进野马优化算法 (Improved Wild Horse Optimizer, IWHO), 并将其运用于 LCLRPSPD 问题。

1 低碳开放式送取货选址路径问题

这里将 LOLRPSPD 描述为, 给定 I 个候选设施点和 J 个需求点, 从 I 个候选设施点中开放指定数量的设施点, 并规划配送路径, 以完成对所有需求点的送取货任务, 使得总成本 (包括设施启用成本、运输路线成本、车辆启用成本、碳排放成本) 最小。为了更适应实际情况, 这里考虑配送车辆在完成配送后不再返回设施点。

1.1 碳排放成本

基于模型的复杂性, 以及在实际情况中难以预测车辆的情况, 这里依照 Xiao 等^[10]的方法将车辆的行驶速度设为匀速, 仅考虑车辆的负载和行驶路程。考虑到车辆在行驶过程中的油耗与负载近似呈线性关系^[12-13], 将其表示为式 (1)。

$$\rho(Q_{ijk}) = (l_k^1 - l_k^0) \frac{Q_{ijk}}{Q_k} + l_k^0 \quad (1)$$

式中: Q_{ijk} 为实时车辆载重; l_k^1 、 l_k^0 分别为车辆满载时和空载时的油耗。通过车辆满载时和空载时的油耗差值乘以载重比例再累加空载油耗, 可求得 $\rho(Q_{ijk})$, 得到实时车辆油耗。常将车辆油耗与碳排放视为等价^[12-13], 故这里计算碳排放成本可采用式 (2)。

$$C_{\text{carbon}} = \omega \mu_k d_{ij} \rho(Q_{ijk}) \quad (2)$$

式中: ω 为单位碳排放价格; μ_k 为燃油转换碳排放系数; d_{ij} 为两点之间的距离。

1.2 符号说明

1) 集合。\$I\$ 为设施点集合, \$J\$ 为需求点集合, \$K\$ 为车辆集合, \$U=I \cup J\$。

2) 常量。\$c_i\$ 为在 \$i\$ 处开放设施点的成本; \$c_{ij}\$ 为 \$i\$ 到 \$j\$ 的运输成本; \$d_{ij}\$ 为 \$i\$ 到 \$j\$ 的距离; \$c_k\$ 为车辆指派成本; \$Q_k\$ 为车辆 \$k\$ 的运载能力; \$Q_i\$ 为设施点 \$i\$ 的最大容量; \$p_j\$ 为需求点 \$j\$ 需要配送的数量; \$q_j\$ 为需求点 \$j\$ 需要取回的数量; \$Q_{ijk}\$ 为车辆 \$k\$ 从 \$i\$ 到 \$j\$ 所装载商品的数量; \$t_{ik}\$ 为车辆 \$k\$ 访问需求点 \$i\$ 的顺序; \$\omega\$ 为单位碳排放价格; \$\mu_k\$ 为燃油转换碳排放系数; \$l_k^0\$ 为车辆 \$k\$ 空载时的油耗; \$l_k^1\$ 为车辆 \$k\$ 满载时的油耗; \$M\$ 为大整数。

3) 决策变量。\$x_{ijk} = \begin{cases} 1 & \text{车辆 } k \text{ 从节点 } i \text{ 驶向节点 } j \\ 0 & \text{其他} \end{cases}\$

\$i, j \in N, i \neq j\$。\$y_i = \begin{cases} 1 & \text{第 } i \text{ 个仓库开放} \\ 0 & \text{其他} \end{cases} \quad i \in I\$。

\$y_{ij} = \begin{cases} 1 & \text{车辆与设施点的指派关系} \\ 0 & \text{其他} \end{cases} \quad i, j \in N\$。

1.3 模型建立

总成本最小值的计算见式 (3), 包括设施点开放成本、运输成本、车辆启用成本和碳排放成本。碳排放的计算见式 (4)。式 (5) 表示每个需求点被访问有且仅有 1 次。式 (6) 表示进入节点的流等于流出节点的流。式 (7) 表示每辆车只能行驶 1 次。式 (8) 表示每个需求点都有唯一指定的设施点为其服务。式 (9) 表示消除子回路。式 (10) 和 (11) 分别表示送取货不超过设施点容量限制。式 (12) 表示车辆装卸平衡约束。式 (13) 表示车辆不能超载。式 (14) 表示车辆运送完成后不返回设施点。式 (15) ~ (17) 为决策变量取值范围。

$$\min z_1 = \sum_{i \in I} c_i y_i + \sum_{i \in U} \sum_{j \in U} \sum_{k \in K} c_{ij} x_{ijk} + \quad (3)$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_k x_{ijk} + \sum_{i \in U} \sum_{j \in U} \sum_{k \in K} \omega \mu_k d_{ij} \rho(Q_{ijk}) x_{ijk}$$

s.t.

$$\rho(Q_{ijk}) = (l_k^1 - l_k^0) \frac{Q_{ijk}}{Q_k} + l_k^0 \quad (4)$$

$$\sum_{i \in U} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in J \quad (5)$$

$$\sum_{j \in U} x_{ijk} - \sum_{j \in U} x_{jik} = 0 \quad \forall i \in U, \forall k \in K \quad (6)$$

$$\sum_{i \in U} \sum_{k \in K} x_{ijk} \leq 1 \quad \forall k \in K \quad (7)$$

$$\sum_{o \in J} x_{io k} + \sum_{o \in U \setminus \{j\}} x_{oj k} \leq 1 + y_{ij} \quad \forall i \in I, \forall j \in J, \forall k \in K \quad (8)$$

$$t_{ik} - t_{jk} + M x_{ijk} \leq M - 1 \quad \forall i \in J, \forall j \in J, \forall k \in K \quad (9)$$

$$\sum_{j \in J} p_j y_{ij} \leq Q_i y_i \quad \forall i \in J \quad (10)$$

$$\sum_{j \in J} q_j y_{ij} \leq Q_i y_i \quad \forall i \in I \quad (11)$$

$$\sum_{j \in J} Q_{jik} - \sum_{j \in J} Q_{ijk} = q_i - p_i \quad \forall i \in I, \forall k \in K \quad (12)$$

$$Q_{ijk} \leq Q_k x_{ijk} \quad \forall i \in U, \forall j \in U, \forall k \in K \quad (13)$$

$$\sum_{k \in K} \sum_{j \in J} x_{jik} = 0 \quad \forall i \in I \quad (14)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in U, \forall j \in U, \forall k \in K \quad (15)$$

$$y_i \in \{0, 1\} \quad \forall i \in I \quad (16)$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (17)$$

上述模型中, 各点位置集合、建设费用、容量及启用成本已知, 车辆启用成本及最大负荷已知, 运输成本已知。送取货需求量和碳排放相关系数由后文拟定。车辆实时载重可由送取货需求计算得到。\$M\$ 为计算过程中的罚成本, 用以筛除不符合约束的路线。

2 野马算法

2.1 基本野马算法

基本野马算法 (Wild Horse Optimizer, WHO) 由 5 个步骤组成: 创建初始种群和挑选领导者 (种马); 马的放牧和交配; 由领导者 (种马) 领导和带领群体; 交流和选拔种马; 保存最佳解决方案。初始种群被分为许多组, \$N\$ 为种群数量, \$P_s\$ 为种马在种群中的比例, \$G\$ 为种马群体, \$G=[N \times P_s]\$, \$(N-G)\$ 为其余环绕种马的马群。放牧行为可用式 (18) 表示。

$$\bar{X}_{i,G}^j = 2Z \cos(2\pi RZ) \times (S_{\text{tallion}}^j - X_{i,G}^j) + S_{\text{tallion}}^j \quad (18)$$

式中: \$\bar{X}_{i,G}^j\$ 为马群 (小马驹或母马) 当前的位置; \$S_{\text{tallion}}^j\$ 为种马位置; \$Z\$ 为适应机制系数; \$R\$ 为 \$[-2, 2]\$ 范围内的随机数。结合 \$\cos\$ 函数使原函数在小范围内扰动, 产生新的 \$\bar{X}_{i,G}^j\$, 表示种群更新后的位置。其中, \$Z\$ 由式 (19) 确定。

$$I_{\text{DX}} = ((\bar{R}_1 < T_{\text{DR}}) = 0); Z = R_2 \Theta I_{\text{DX}} + \bar{R}_3 \Theta (\sim I_{\text{DX}}) \quad (19)$$

式中: \$\bar{R}_1\$、\$\bar{R}_3\$ 为 \$[0, 1]\$ 空间内均匀分布的随机向量; \$R_2\$ 为 \$[0, 1]\$ 之间的随机数; \$I_{\text{DX}}\$ 为维度与 \$\bar{R}_1\$ 相同的向量, 其元素值与 \$\bar{R}_1\$ 有关。

若 \$\bar{R}_1\$ 的第 1 个元素小于 \$T_{\text{DR}}\$, 则 \$I_{\text{DX}}\$ 的第 1 个元素为 0, 反之为 1, 依次类推; \$T_{\text{DR}}\$ 为依迭代次数从 1 线性递减到 0 的自适应因子; \$\Theta\$ 为点乘; \$\sim\$ 为二进制取反。交配行为由临时组中来自不同种群的小马驹产生, 见式 (20)。

$$X_{G,k}^p = \text{Crossover}(X_{G,i}^q, X_{G,j}^z) \quad i \neq j \neq k, p=q=\text{end} \quad (20)$$

Crossover = Mean

式中: \$X_{G,k}^p\$ 为种群 \$k\$ 中个体 \$p\$ 离群后再次进入种群 \$k\$ 的个体位置; \$X_{G,i}^q\$、\$X_{G,j}^z\$ 分别表示来自不同组 (\$i\$

组和 j 组) 的父代 q 和母代 z 。

之后, 种马带领种群前往合适的水洼, 但若该水洼已被另一个种群占领, 则后来的种马只能找寻其他水洼。具体计算见式 (21)。

$$\overline{S_{\text{tallion}G_i}} = \begin{cases} 2Z \cos(2\pi RZ) \times (W_H - S_{\text{tallion}G_i}) + W_H & R_3 > 0.5 \\ 2Z \cos(2\pi RZ) \times (W_H - S_{\text{tallion}G_i}) - W_H & R_3 \leq 0.5 \end{cases} \quad (21)$$

式中: $\overline{S_{\text{tallion}G_i}}$ 为 i 组种马的新位置; W_H 为水洼的位置; $S_{\text{tallion}G_i}$ 为 i 组种马现在的位置。

再后, 当种马适应度被其他成员超越时, 则执行种马交换, 见式 (22)。

$$S_{\text{tallion}G_i} = \begin{cases} X_{G_i} & \text{if } \cos t(X_{G_i}) < \cos t(S_{\text{tallion}G_i}) \\ S_{\text{tallion}G_i} & \text{if } \cos t(X_{G_i}) > \cos t(S_{\text{tallion}G_i}) \end{cases} \quad (22)$$

2.2 编码及解码方式

采用实数编码方式, 每个基因的取值为 0~1 内的实数。重要的是解码方式, 在计算适应度前解码。解码方式如图 1 所示, 给定具有可选设施点 5 个、需求点 20 个、备选车辆 7 辆的 LRP, 其染色体维度为 $5+7+20=32$ 。

1) 前 5 个基因用于选择设施点 i 进行建设, 这里选择小于 0.5 的位置, 即该例中的 i_4 和 i_5 。

2) 之后的 7 个基因代表备选车辆, 分别乘以可建设设施点 i 的数目 2(从步骤 1 得出), 并向上取整,

得到备选车辆所服务设施点的编号。需要说明的是, 这里求出的编号 2 代表建设的第 2 个设施点 i , 即 i_5 , 编号 1 表示 i_4 , 如图 2 所示。

3) 剩余的 20 个基因代表需求点, 用于计算需求点所选择的车辆及运输顺序。同理, 将其乘以备选车辆数量 7, 并向上取整, 得到每个需求点所选车辆编号。若此步骤未出现某辆候选车辆编号, 则此车辆停止使用。之后分别按基因排序, 可得车辆服务需求点的顺序。以第 2 辆车为例, 共有编号 5、9、10、11 需要 2 号车服务, 如图 3 所示。

对以上结果进行升序排列, 可得到顺序 $2 \rightarrow 3 \rightarrow 1 \rightarrow 4$, 即 $j_9 \rightarrow j_{10} \rightarrow j_5 \rightarrow j_{11}$, 所以第 2 辆车的顺序为 $i_4 \rightarrow j_9 \rightarrow j_{10} \rightarrow j_5 \rightarrow j_{11}$ 。

通过此种编码方式, 仓库开放与否、客户点服务顺序及归属清晰可见。另外, 此编码考虑了实际情况中车辆运输能力的差异性, 更加符合数学模型和实际情况。

2.3 改进野马算法

这里详细介绍改进野马算法 (IWHO) 的改进策略。为了增强基本算法的全局搜索能力、提升初始化种群多样性, 改善后期难以逃脱局部最优等问题, 进行了多次实验。这里改进了 6 种方法, 包括引入哈尔顿序列生成初始种群、非线性进化参数、模拟二进制交叉、多项式变异、精英保留策略、多次失败后个体重新初始化。

0.600 12	0.514 600	0.660 142	0.305 701	0.042 197	0.603 829	0.279 845	0.172 094
0.328 808	0.791 340	0.635 294	0.613 526	0.079 551	0.476 317	0.647 337	0.781 168
0.216 452	0.969 519	0.325 509	0.527 790	0.146 607	0.158 408	0.272 457	0.597 100
0.069 189	0.378 727	0.858 150	0.549 523	0.847 243	0.457 617	0.656 979	0.750 517

图 1 完整染色体实例
Fig.1 Complete chromosome instance



图 2 备选车辆基因选择设施点流程
Fig.2 Alternative vehicle gene selection facility point process

每个需求点对应服务车辆									
1	4	5	6	2	7	3	4	2	2
2	5	1	3	7	4	6	4	5	6

2号车服务的需求点基因			
0.216452	0.146607	0.158408	0.272457

图 3 需求点基因解码逻辑

Fig.3 Demand point gene decoding logic diagram

2.3.1 初始种群的生成

算法的种群初始化阶段会极大地影响算法的后期处理。基本的 WHO 是先根据种群数量选择等量种马, 再在其他所有马驹里随机选择个体跟随指定的种马行动。这样会导致前期种群的混乱。例如, 存在一个较好的马驹个体, 在周围有优秀种马的情况下, 因过分随机而被分配给遥远的种马, 使得该马驹在迭代初期并未向有利个体靠近。在多次重复实验过程中发现, 针对 LRP, 基本 WHO 算法这种对于前期种群过于“混乱”的处理方式, 对整体迭代有着不良影响。Halton 序列是一种更接近于均匀分配的伪随机。经大量反复实验后可以得出结论, 针对 LRP 引入 Halton 序列, 可以降低算法开局迭代中的种群混乱程度, 加快全局搜索。

2.3.2 非线性进化概率因子

在原始算法中, 放牧行为的进化概率参数 (T_{DR}) 按迭代次数线性递减, 见式 (23)。

$$T_{DR} = 1 - \frac{N_{iter}}{N_{iterations}} \quad (23)$$

式中: N_{iter} 为当前迭代次数; $N_{iterations}$ 为总迭代次数。

在处理 LRP 时, 该算法到后期基本收敛, 导致后期进化幅度过大, 难以逃脱局部最优。为了更好地平衡算法全局和局部搜索能力, 防止算法在中后期过早进入停滞状态, 参考 logsig 函数, 改进原有的线性进化概率因子, 使之平滑过渡。改进后 T_{DR} 的计算见式 (24)。

$$T_{DR} = \frac{1}{1 + e^{-(0.5 - \frac{N_{iter}}{N_{iterations}}) * 5}} \quad (24)$$

2.3.3 模拟二进制交叉算子

原算法在第 2 部分交配的过程中, 使用单点/多点交叉。在实验过程中发现, 当数据维度过大时效果很差, 故使用模拟二进制交叉算子 (Simulated binary crossover, SBX)。SBX 算子是一种模拟单点二进制交叉的交叉算子。假设 2 个父代个体 $x^1(x_1^1, \dots, x_n^1)$ 和 $x^2(x_1^2, \dots, x_n^2)$, 则 2 个后代个体 $c^1(c_1^1, \dots, c_n^1)$ 和 $c^2(c_1^2, \dots, c_n^2)$ 通过式 (25) 得到。

$$\begin{cases} c_i^1 = 0.5 \times [(1 + \beta) \cdot x_i^1 + (1 - \beta) \cdot x_i^2] \\ c_i^2 = 0.5 \times [(1 - \beta) \cdot x_i^1 + (1 + \beta) \cdot x_i^2] \end{cases} \quad (25)$$

其中, β 由分布因子 η 按照公式动态随机决定, 见式 (26)。

$$\beta = \begin{cases} (\text{rand} \times 2)^{1/(1+\eta)} & \text{rand} \leq 0.5 \\ 1 / (2 - \text{rand} \times 2)^{1/(1+\eta)} & \text{otherwise} \end{cases} \quad (26)$$

η 为自定义参数, η 越大, 则产生的后代逼近父代个体的概率越大。SBX 算子在使用实数编码的进化算法中, 针对局部搜索的表现优越, 且处理高维目标优化问题对于个体空间稀疏性的效果良好, 十分适用于 LRP。

2.3.4 多项式变异算子

在原算法交叉环节后加入变异操作, 即使用多项式变异 (Polynomial Mutation Operator, PMO) 算子对当前解施加扰动, 并以一定概率接受劣化解, 且接受劣化解的概率随时间的推移而降低。一方面, 可以在算法前期维持解的多样性, 以加强全局搜索能力。另一方面, 可以使算法后期变异产生的新解逐渐逼近当前解, 提升局部随机搜索能力。多项式变异算子形式见式 (27) ~ (28)。

$$v_{k+1} = v_k + \delta \cdot (u_k - l_k) \quad (27)$$

$$\delta = \begin{cases} [2u + (1 - 2u)(1 - \delta_1)^{\eta_m}]^{\frac{1}{\eta_m+1}} - 1 & u \leq 0.5 \\ 1 - [2(1 - u) + 2(u - 0.5)(1 - \eta_2)^{\eta_m}]^{\frac{1}{\eta_m+1}} & u > 0.5 \end{cases} \quad (28)$$

式中: v_k 为变异前个体位置; v_{k+1} 为变异后个体位置; u_k 为位置上限; l_k 为位置下限; u 为 $[0, 1]$ 上的随机数; η_m 为分布指数; $\delta_1 = (v_k - l_k) / (u_k - l_k)$; $\delta_2 = (u_k - v_k) / (u_k - l_k)$ 。

2.3.5 精英保留策略

为了避免交叉及变异步骤丢失父代种群中的精英个体, 采取精英保留策略存储运算过程中的精英个体。精英个体是种群进化中搜索到的具有最优适应度的个体, 它包含目前最好的基因数据。精英保留策略的核心思想是把种群进化过程中出现的精英个体复制到下一代中。在变异后进行精英保留操作, 通过合并交叉和变异的种群, 并与原种群对比, 选择最优的保留下来, 有效地提高了算法的收敛能力。

2.3.6 连续失败重新初始化

原始野马算法极易陷入局部最优, 为了改善这种情况, 在算法中设定最大失败统计数 N_{fail} (设定为迭代次数的 $1/10$)。当算法连续运行 N_{fail} 都未找到更好的解时, 说明局部收敛。此时对个体重新初始化, 计算个体适应度, 并进行迭代。经大量实验证明, 该方法对防止算法过快陷入局部最优有效。

2.4 改进野马算法流程描述

改进野马算法 (IWHO) 求解低碳开放式送取货选址路径问题 (LOLRSPD) 步骤如下。

1) 初始化算法参数。包括种群规模 N 、种马比

例 P_S 、最大迭代次数 T 、交配概率 P_C 等。

2) 解码。计算种群个体适应度, 通过哈尔顿序列初始化种群, 并选取种马。

3) 根据式 (24) 计算 T_{DR} , 根据式 (18) ~ (19) 进行放牧行为。

4) 先根据式 (20) 进行交配行为, 再根据式 (25) ~ (26) 进行二进制交叉, 根据式 (27) ~ (28) 进行多项式变异, 然后解码, 并按精英保留策略将优秀个体保留。

5) 根据式 (21), 种马选择新的水洼(目标区域)。

6) 解码。更新种群适应度, 若种群中存在适应度优于种马个体, 则根据式 (22) 更换种马。

7) 若连续失败, 则重新初始化种群, 转到步骤 2)。

8) 判断是否达到最大迭代次数, 若是, 终止循环, 否则转到步骤 3)。

9) 满足算法终止条件, 输出最优解。

2.5 时间复杂性分析

原始野马算法时间复杂度参考 Li 等^[14]的方法, 设种群的大小为 N , 最大迭代次数为 T , 个体维度为 n , 则初始化种群, 并选择种马阶段的时间复杂度 $O_1(Nn)$ 、放牧阶段时间复杂度 $O_2(NTn)$ 、交配阶段时间复杂度 $O_3(NTn)$, 以及领导者阶段时间复杂度 $O_4(NTn)$ 。故原始野马算法时间复杂度 $O(NTn) = O_1(Nn) + O_2(NTn) + O_3(NTn) + O_4(NTn)$ 。

文中对野马算法的变动有 6 点。首先, 在初始种群阶段引入 Halton 序列, 替代原有随机序列, 然后保持时间复杂度不变, 仍为 $O_1(Nn)$ 。在放牧阶段改进了进化概率参数 T_{DR} , 虽然增加了运算量, 但未产生额外的循环, 故时间复杂度仍为 $O_2(NTn)$ 。交配阶段改变较大, 改进了交叉方式, 使用多项式交叉, 又增加了多项式变异和精英保留策略。对于交叉与变异, 依旧只增加运算量, 未产生额外的循环。在精英保留步骤中, 多使用了 1 个循环, 影响较大, 故改进后的交配阶段时间复杂度为 $O_3(NTn^2)$ 。领导者阶段不变, 时间复杂度仍为 $O_4(NTn)$ 。最后新增连续失败重新初始化阶段, 时间复杂度应为 $O_5(Nn)$ 。故改进野马算法的时间复杂度 $O(NTn^2) = O_1(Nn) + O_2(NTn) + O_3(NTn^2) + O_4(NTn) + O_5(Nn)$ 。

3 算例分析

为了评估改进野马算法求解 LOLRPSPD 的性能, 使用 Matlab R2021a 实现该算法, 在 64 位 Windows10 专业版系统下配备 AMD Ryzen 5 3600X 3.80 GHz 处理器和 16 G 内存的主机上进行大量数据实验。为了验证 IWHO 求解 LOLRPSPD 模型的有效性, 在 Prins 等^[15]提出的标准 CLRP 算例数据上进行了相应改进, 利用 AM 分离法^[7]生成需求点的送取货需求, 即令客户点送货需求为原需求, 取货需求 p_i 的

计算见式 (29)。

$$p_i = \begin{cases} (1+\lambda)q_i & i \text{ 为奇数} \\ (1-\lambda)q_i & i \text{ 为偶数} \end{cases} \quad (29)$$

式中: q_i 为算例中的原始需求; λ 取 0.7。

对于碳排放相关系数, 考察了现阶段国内市场情况, 并根据一般运费比例, 将单位碳排放价格 ω 定为 10, 燃油转换碳排放系数 μ_k 定为 2.6, 空载单位距离油耗 l_k^0 为 0.75, 满载单位距离油耗 l_k^1 为 1.5。给出了标准算例库中的 6 组算例, 见表 1。

表 1 算例数据
Tab.1 Example data

算例 编号	名称	设施点 数量	需求点 数量	车辆启用 成本	车辆 容量
1	coord20-5-1	5	20	1 000	70
2	coord20-5-1b	5	20	1 000	70
3	coord50-5-1	5	50	1 000	70
4	coord50-5-1b	5	50	1 000	150
5	coord100-5-1b	5	100	1 000	150
6	coord100-10-1	10	100	1 000	150

3.1 算法参数设置

为了得到优秀的参数组合, 分别对算例 1 和算例 3 进行了分析。通过比较每个算例最优目标函数值和平均运行时间, 对种群规模 N 、种马数量 G 和迭代次数进行了灵敏度分析。在研究其中 1 个参数的性能时, 保持其他参数不变。表 2~4 分别为 3 种参数在 6 组不同取值情况下对 2 组不同算例的平均运行结果,

表 2 种群规模对解的影响
Tab.2 Influence of population size on the solution

种群规模	总成本		运行时间/s	
	20-5	50-5	20-5	50-5
20	47 739	111 102	24.0	44.6
40	47 201	106 064	23.0	44.6
60	42 538	104 550	23.3	45.5
80	48 683	94 758	23.7	44.2
100	45 906	98 291	23.5	43
120	51 246	106 436	23.7	45.5

表 3 种马数量对解的影响
Tab.3 Influence of the number of stallions on the solution

种马数量	总成本		运行时间/s	
	20-5	50-5	20-5	50-5
6	50 855	111 179	8.2	15.1
12	46 569	91 485	23.8	44.0
18	52 230	102 329	49.7	97.6
24	43 328	100 107	84.0	169.4
30	45 255	91 085	128.4	256.6
36	42 588	94 683	185.2	362.9

黑体表示最优结果。对应的变化曲线如图 4 所示。不难看出, 种群规模 N 的改变对运行时间的影响不大, 但当 N 过大或过小时, IWHO 很容易陷入局部最优, 从而产生较差的平均目标值。种马数量 G 的改变对运行时间的影响很大, 运行时间随着种马数量的提升显著延长。在种马数量为 36 时, 运行时间甚至超过种马数量为 12 时的 8 倍。对总成本的影响可以从折线图看出, 算例 1 和算例 3 均在种马数量为 12 时存在明显的低谷, 在结合了适当运行时间的情况下, 这里选择种马数量 12 作为实验标准。迭代次数对解的影响最明显, 在基本不影响运行时间的前提下, 2 组算例均在迭代次数为 40 时取到最优成本。综上所述, 选取种群规模 $N=60$ 、种马数量 $G=12$ 、迭代次数 40 作为实验标准。

表 4 迭代次数对解的影响				
Tab.4 Influence of iteration times on the solution				
迭代次数	总成本		运行时间/s	
	20-5	50-5	20-5	50-5
20	50 408	93 972	24.1	45.2
30	52 964	103 151	24.1	46.0
40	46 771	92 780	23.7	44.4
50	48 223	109 026	23.6	46.8
60	47 492	112 106	23.7	44.0
70	49 079	112 106	23.1	44.8

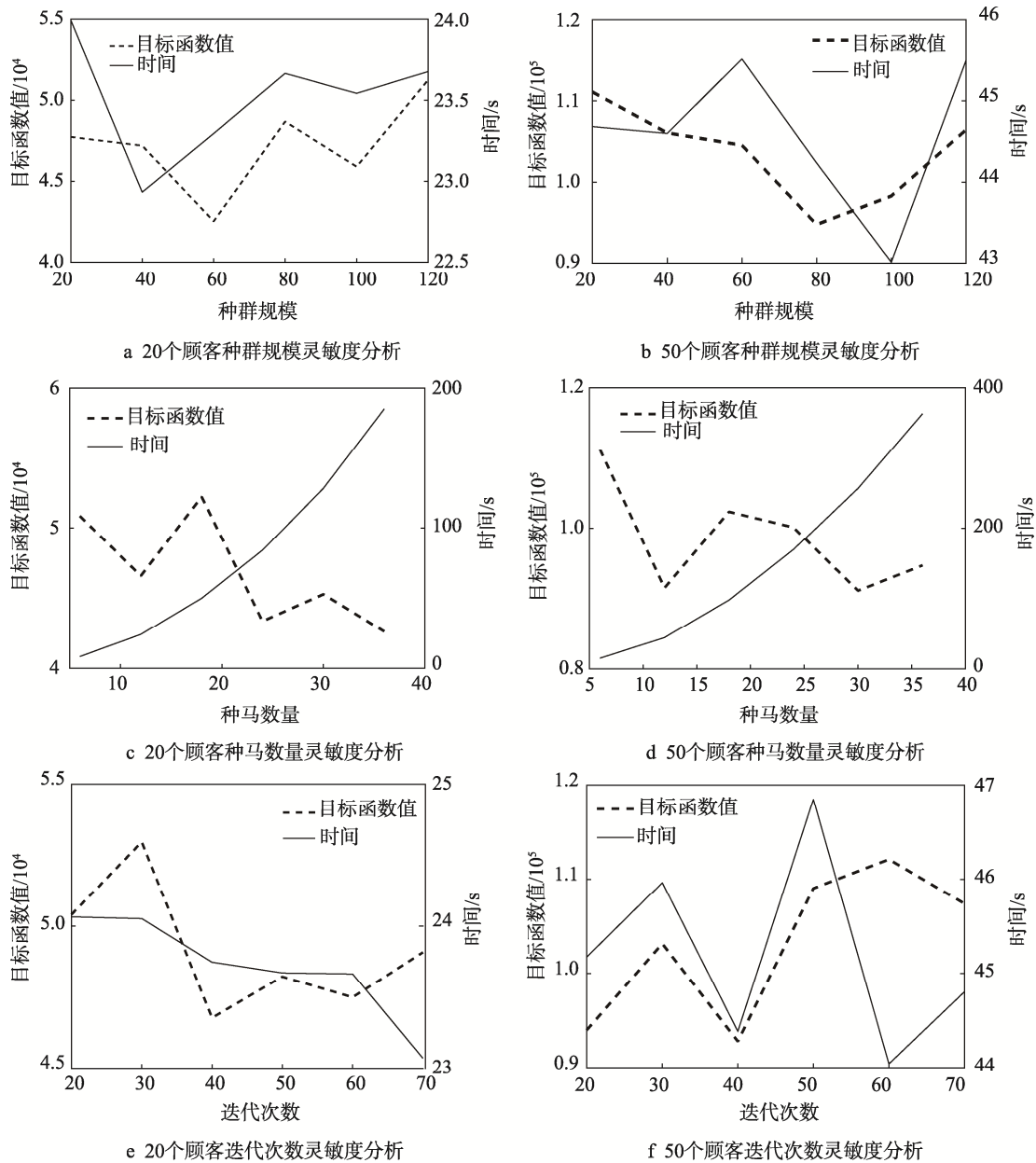


图 4 参数灵敏度分析
Fig.4 Parameter sensitivity analysis

3.2 算法性能分析

为了检验 IWHO 算法的性能, 针对以上 6 个算例进行求解, 并与原始野马算法 (WHO)、粒子群优化算法 (Particle Swarm Optimization, PSO)、模拟退火算法 (Simulated Annealing, SA)、遗传算法 (Genetic Algorithm, GA) 进行了对比。实验对每个算例求解 20 次, 记录最优解及所用时间, 求解结果如表 5 所示。

实验结果表明, 针对中型 (编号 1~2)、大型 (编号 3~6) 算例, 文中所提改进野马算法相较于原始野马算法具有显著优势。从运行结果来看, 由于 LOLRSPD 问题的复杂性, 原始野马算法在实验过程中极易过早陷入局部收敛, 难以获得最优结果。IWHO 在 2 组中型算例中都取得了已知最优解, 说明改进后非线性进化概率因子及连续失败多次重新初始化步骤均取得了十分理想的结果。从运行稳定性来看, IWHO 针对中型算例每次都能求得已知最优解, 且迭代次数不多。原始野马算法不论在中型算例还是大型算例上的结果都难令人满意, 在中型算例上, 即使迭代 20 次都无法求得已知最优解, 且结果平均偏差在 15%以上。对比分析后不难得出, 改进后运用 Halton 序列种群初始化配合多次失败重新初始化产生了良好效果。从运算速度来看, 虽然 IWHO 每次迭代的时间约为原始 WHO 的 3 倍, 但得出合适解的总时间更短。由于 WHO 在处理 LOLRSPD 问题上总是过早收敛, 很难统计有效时间, 因此对于 2 组中型算例, 这里在规定运行时间的前提下设置了合适的迭代次数。实验结果基本证明文中所提的二进制交叉算子 SBX、多项式变异算子 (PMO) 及精英保留策略较大程度地改善了原始 WHO 的全局搜索能力, 使得 IWHO 可以适应 LRP。

与其他 3 个经典算法相比, 遗传算法 GA 和粒子群算法 PSO 都较适合求解 LOLRSPD 问题。这可能是因为算法本身的全局搜索能力较强, 且收敛速度较慢。模拟退火算法 SA 与原始野马算法 WHO 一样, 虽然局部搜索能力较强, 但全局搜索能力较弱、收敛速度较快, 不太适合求解 LRP 类问题。针对 20 个需求点的中型算例, 改进野马算法 IWHO 对比 PSO 和 GA 在时间上具有优势。针对 50 个需求点的大型算例, IWHO 可以一次求得已知最优解, 但求解时间弱于 GA。针对 100 个需求点的大型算例, IWHO 会过早陷入局部最优, 难以求出超过 GA 的结果, 但胜过 PSO。

从实验结果推断, 求解 LOLRSPD 问题更需要算法的广度, 即全局搜索能力。虽然原始野马算法 WHO 拥有不错的局部搜索能力, 但初始化种群不稳定、全局搜索能力弱、收敛快, 不适合求解 LOLRSPD 问题。文中提出的改进野马算法 IWHO 在保留原算法优势的同时, 弥补了其劣势, 处理小型 LRP 问题具有一定优势。

3.3 IWHO 有效性评估

为了进一步验证 IWHO 在求解 LRP 上的有效性, 这里用 IWHO 求解了 6 组标准 CLRP 数据库中的算例, 将其求解结果与 GRASP^[15]、MAPW^[16] 和 2phase-HGTS^[17] 的求解结果进行对比分析。其中, Best 是算例已知最优解。GRASP、MAPW 和 2phase-HGTS 由 C++ 编写, 运行环境参见文献[15-17]。IWHO 由 Matlab 编写。

由表 6 可知, IWHO 可以取得前 5 个中型算例的最优解, 证明文中提出的改进野马算法 IWHO 在处理 LRP 上可信。

表 5 5 种算法求解 LOLRSPD 算例对比
Tab.5 Comparison of five algorithms for solving LOLRSPD examples

编号	名称	IWHO		WHO		PSO		SA		GA	
		总成本	运行时间/s	总成本	运行时间/s	总成本	运行时间/s	总成本	运行时间/s	总成本	运行时间/s
1	20-5-1	40 788	23	47 145	40	40 788	28	43 125	24	40 788	27
2	20-5-1b	31 149	23	37 675	40	31 149	23	33 670	24	31 149	27
3	50-5-1	89 321	65	138 421	120	90 268	78	117 837	58	88 867	60
4	50-5-1b	65 837	66	80 900	120	68 598	70	72 534	53	65 837	55
5	100-5-1b	160 875	162	209 288	153	189 224	274	253 123	233	154 385	262
6	100-10-1	184 088	281	245 513	318	198 233	374	284 124	344	175 923	351

表 6 标准算例的求解结果
Tab.6 Solution results of standard examples

算例	Best	GRASP		MAPW		2phase-HGTS		IWHO	
		最优解	时间/s	最优解	时间/s	最优解	时间/s	最优解	时间/s
Gaskell67-21x5	424.9	429.6	0.2	424.9	0.3	424.9	6	424.9	17.5
Gaskell67-22x5	585.1	585.1	0.2	611.8	0.3	585.1	9	585.1	18.4
Gaskell67-29x5	512.1	515.1	0.4	512.1	0.8	512.1	11	512.1	26.8
Gaskell67-32x5	562.2	571.9	0.6	571.9	0.8	562.2	40	562.2	34.7
Gaskell67-32x5-2	504.3	504.3	0.5	534.7	1.0	504.3	22	504.3	26.2
Gaskell67-36x5	460.4	460.4	0.8	485.4	1.4	460.4	39	479.5	48.1

4 结语

在开放式送取货选址-路径问题模型上加入碳排放成本,以适应目前的低碳背景,并使用改进的野马算法进行求解。通过引入哈尔顿序列生成初始种群、非线性进化参数、模拟二进制交叉、多项式变异、精英保留策略和多次失败后个体重新初始化等,有效改善了原始算法针对 LRP 存在的初始种群分布不均、全局搜索能力弱和容易陷入局部最优等问题。算例分析结果表明,文中所提的改进 IWHO 算法对于 LCLRPSPD 问题具有良好的优化性能。

为了进一步加强该问题的现实意义,人文关怀是不可或缺的,因此后续研究将构建多目标模型,考虑顾客满意度及配送员工作量,并设计其相应的 IWHO 求解算法。

参考文献:

[1] COOPER L. The Transportation-Location Problem[J]. Operations Research, 1972, 20(1): 94-108.

[2] KARAOGLAN I, ALTIPARMAK F, KARA I, et al. Formulations for Location-Routing Problem with Simultaneous Pickup and Delivery[E/OL]. 2009. http://w3.gazi.edu.tr/~fulyaal/Papers/LRPSPD_MIP Formulations.

[3] KARAOGLAN I, ALTIPARMAK F, KARA I, et al. A Branch and Cut Algorithm for the Location-Routing Problem with Simultaneous Pickup and Delivery[J]. European Journal of Operational Research, 2011, 211(2): 318-332.

[4] KARAOGLAN I, ALTIPARMAK F, KARA I, et al. The Location-Routing Problem with Simultaneous Pickup and Delivery: Formulations and a Heuristic Approach[J].

Omega, 2012, 40(4): 465-477.

[5] WANG X F. An Location-Routing Problem with Simultaneous Pickup and Delivery in Urban-Rural Dual-Directions Logistics Network[J]. Advanced Materials Research, 2013, 756/757/758/759: 3423-3429.

[6] 刘亚平, 张惠珍, 张莉, 等. 带时间窗同时送取货选址路径问题及其烟花算法求解[J]. 计算机应用, 2022, 42(7): 2292-2300.

LIU Y P, ZHANG H Z, ZHANG L, et al. Fireworks Algorithm for Location-Routing Problem of Simultaneous Pickup and Delivery with Time Window[J]. Journal of Computer Applications, 2022, 42(7): 2292-2300.

[7] 刘冬, 张惠珍, 刘亚平, 等. 改进蘑菇算法求解开放式同时送取货选址-路径问题[J]. 控制工程, 2023, 30(10): 1801-1811.

LIU D, ZHANG H Z, LIU Y P, et al. An Improved Mushroom Algorithm for Solving the Open Location-Routing Problem with Simultaneous Pickup and Delivery[J]. Control Engineering of China, 2023, 30(10): 1801-1811.

[8] BARTH M, YOUNGLOVE T, SCORA G. Development of a Heavy-Duty Diesel Modal Emissions and Fuel Consumption Model, UCB-ITS-PRR-2005-1[R]. California PATH Research Report, 2005.

[9] ZHANG J H, ZHAO Y X, XUE W L, et al. Vehicle Routing Problem with Fuel Consumption and Carbon Emission[J]. International Journal of Production Economics, 2015, 170: 234-242.

[10] XIAO Y Y, ZHAO Q H, KAKU I, et al. Development of a Fuel Consumption Optimization Model for the Capacitated Vehicle Routing Problem[J]. Computers & Op-

- erations Research, 2012, 39(7): 1419-1431.
- [11] NARUEI I, KEYNIA F. Wild Horse Optimizer: A New Meta-Heuristic Algorithm for Solving Engineering Optimization Problems[J]. Engineering with Computers, 2022, 38(4): 3025-3056.
- [12] 张春苗, 赵燕伟, 张景玲, 等. 低碳定位——车辆路径问题[J]. 计算机集成制造系统, 2017, 23(12): 2768-2777.
- ZHANG C M, ZHAO Y W, ZHANG J L, et al. Location and Routing Problem with Minimizing Carbon[J]. Computer Integrated Manufacturing Systems, 2017, 23(12): 2768-2777.
- [13] 王舜, 赵燕伟, 冷龙龙, 等. 基于蚁群选择超启发算法的低碳选址-路径问题[J]. 计算机集成制造系统, 2020, 26(6): 1702-1716.
- WANG S, ZHAO Y W, LENG L L, et al. Low Carbon Location-Routing Problem Based on Evolutionary Hyper-Heuristic Algorithm of Ant Colony Selection Mechanism[J]. Computer Integrated Manufacturing Systems, 2020, 26(6): 1702-1716.
- [14] LI Y C, YUAN Q Y, HAN M X, et al. Hybrid Multi-Strategy Improved Wild Horse Optimizer[J]. Advanced Intelligent Systems, 2022, 4(10): 2200097.
- [15] PRINS C, PRODHON C, CALVO R W. Solving the Capacitated Location-Routing Problem by a GRASP Complemented by a Learning Process and a Path Re-linking[J]. 4OR, 2006, 4(3): 221-238.
- [16] PRINS C, PRODHON C, CALVO R W. A Memetic Algorithm with Population Management (MA|PM) for the Capacitated Location-Routing Problem[M]// Evolutionary Computation in Combinatorial Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 183-194.
- [17] ESCOBAR J W, LINFATI R, TOTH P. A Two-Phase Hybrid Heuristic Algorithm for the Capacitated Location-Routing Problem[J]. Computers & Operations Research, 2013, 40(1): 70-79.