计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(11):83−94 [doi: 10.15888/j.cnki.csa.009305]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

离散灰狼优化算法求解 VRPSPDTW 问题①

陈　凯, 邓志良, 龚毅光

(南京信息工程大学 自动化学院, 南京 210044)
通信作者: 陈　凯, E-mail: 347052630@qq.com

摘　要: 本文针对带时间窗约束的同时送取货车辆路径问题, 建立了以总配送距离最小化为目标的数学模型. 根据模型的特征, 在保留灰狼算法 (GWO) 搜索机制的基础上, 提出了离散灰狼优化算法 (DGWO) 进行求解. 采用多种策略构建种群的初始解, 并允许出现不可行解, 扩大种群的搜索区域; 引入带评分策略的邻域搜索策略, 调整每种算子的概率, 使算法选择优化效果更好的算子; 使用移除-插入机制, 对优质解区域进行探索, 加速种群的收敛. 在仿真实验中对标准数据集进行了测试, 将实验结果和 p-SA 算法、DCS 算法、VNS-BSTS 算法和 SA-ALNS 算法进行了对比, 实验表明 DGWO 算法能有效地解决带时间窗约束的同时送取货车辆路径问题.

关键词: 车辆路径问题; 同时送取货; 灰狼算法; 时间窗; 邻域搜索

引用格式: 陈凯,邓志良,龚毅光.离散灰狼优化算法求解 VRPSPDTW 问题.计算机系统应用,2023,32(11):83–94. http://www.c-s-a.org.cn/1003-3254/9305.html

Discrete Grey Wolf Optimization Algorithm for VRPSPDTW Problem

CHEN Kai, DENG Zhi-Liang, GONG Yi-Guang

(School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044 , China)

Abstract: In this study, a mathematical model aiming at minimizing the total distribution distance is established for the vehicle routing problem of simultaneous delivery and pickup with time window constraints. According to the characteristics of the model, a discrete grey wolf optimization (DGWO) algorithm is proposed to solve the problem on the basis of preserving the search mechanism of the grey wolf optimization (GWO) algorithm. Multiple strategies are adopted to construct the initial solution of the population, and the unfeasible solution is allowed to expand the search area of the population; the neighborhood search strategy with scoring strategy is introduced to adjust the probability of each operator so that the algorithm can select the operator with better optimization effect; the deletion-insertion mechanism is used to explore the high-quality solution region and accelerate the convergence of the population. The standard data set is tested in the simulation experiment, and the experimental results are compared with the p-SA algorithm, DCS algorithm, VNS-BSTS algorithm, and SA-ALNS algorithm. The experiment shows that the DGWO algorithm can effectively solve the vehicle routing problem of simultaneous delivery and pickup with time window constraints.

Key words: vehicle routing problem (VRP); simultaneously deliver and pick up the goods; grey wolf optimization (GWO) algorithm; time window; neighborhood search

当前, 我国经济已从高速增长阶段转向高质量发展阶段, 正处在转变发展方式、优化经济结构、转换增长动力的攻关期. 迫切要求物流业以供给侧结构性改革为主线, 推动物流产业质量变革、效率变革和动

① 基金项目: 国家重点研发计划 (2018YFC1405703)
收稿时间: 2023-05-09; 修改时间: 2023-06-06; 采用时间: 2023-06-26; csa 在线出版时间: 2023-09-19
CNKI 网络首发时间: 2023-10-25

力变革, 实现高质量发展. 物流业的核心部分为车辆配送, 而车辆路径规划问题 (vehicle routing problem, VRP)[1]. 是其中重要的一部分. 传统的 VRP 是指由配送中心向一系列客户交付货物, 并规划合理的行驶路线, 目标是满足客户的需求. 随着环保意识的增强和物流双向流通的发展, 送取货的车辆路径问题也不断得到重视. 为了提升企业的服务质量和客户的满意度, 还需要将客户能接受的服务时间考虑在内. 因此, 研究带时间窗和同时送取货的车辆路径问题 (VRPSPDTW) 具有重要的理论和实践意义.

VRPSPDTW 问题是 VRP 的扩展, 已被证明是一个 NP-hard 问题[2]. VRPSPDTW 的求解算法主要分为两类, 一类是运筹学算法, 包括动态规划、分支定价等[3,4]. 主要采用几何分析和线性代数作为基本方法, 利用目标函数和约束项结构信息来求解问题. Angelelli 等[4] 采用了分支定价法得到了小规模问题的最优解. 当数据规模变大时, VRPSPDTW 的求解会更加复杂, 甚至可能需要数天才能得到满意解. 另一类为智能算法[5−9], 包括遗传算法、布谷鸟算法、粒子群算法等. 这类算法不依赖问题的结构, 通过对解的编码解码处理约束, 不断生成新的可行解. 往往只需要较短的时间, 便可得到不同规模问题的满意解决方案. 因此, 使用智能算法来解决 VRPSPDTW 问题, 是非常必要且合理的. Wang 等[10] 设计了协同遗传算法完成了对 VRPSPDTW 问题的求解, 并设计了 64 组 VRPSPDTW 测试算例. 黄务兰等[11] 改进了人工鱼群算法, 优化了算法的搜索能力, 降低了计算复杂度, 并通过实验验证了算法的可行性. 王超等[12] 提出了布谷鸟算法离散化来求解 VRPSPDTW 问题, 并刷新了国际最优解. Belgin 等[13] 通过将局部搜索算法与变邻域下降搜索相结合, 求解了 VRPSPDTW 问题. Lagos 等[14] 采用改进的粒子群算法完成了 VRPSPDTW 问题的寻优. Wang 等[15] 在 VRPSPDTW 的基础上添加了多配送中心限制, 并采用聚类算法得到满意解. 根据以上文献可知, 1) 大多数求解 VRP 的方法仍然是元启发式算法; 2) 在算法的局部搜索过程中, 大多采用固定的顺序执行插入操作, 难以实现种群的深入搜索.

灰狼算法 (grey wolf optimization, GWO) 是澳大利亚学者 Mirjalili 等[16] 提出的一种群体智能优化算法. 该算法是根据灰狼的捕食行为而开发的一种优化算法, 它具有收敛性强、参数少、易实现等特点. 然而 GWO 最初是为了优化连续问题而设计的, 在求解连续优化问题方面有着广泛的应用, 但在车辆路径优化等离散问题上的应用较少. 为了能将 GWO 应用于离散优化问题, 黄戈文等[17] 设计了一种采用灰狼整数编码的自适应灰狼更新策略来求解 CVRP 问题. 姜天华[18] 针对车间调度问题, 设计了一种离散灰狼算法进行求解, 该算法采用两段式编码, 建立 GWO 连续空间与离散空间的映射关系. 顾九春等[19] 针对作业车间节能调度问题, 设计了一种多目标离散灰狼算法进行求解, 该算法基于速度-工序的离散整数编码方式, 并引入跟踪和搜寻模式. 综上所述, 现有文献采用 GWO 解决离散组合优化问题 (特别是 VRP) 的研究非常有限. 除此之外, 现有的文献针对离散优化问题所设计的离散灰狼优化算法存在局限性, 主要体现在算法对 GWO 的原有机制过于保留, 在个体位置更新时, 算法的离散化不够彻底, 影响算法的运行效率.

因此, 本文构建了以最小化车辆行驶距离为优化目标的 VRPSPDTW 数学模型, 并提出了一种离散灰狼优化算法 (discrete grey wolf optimization, DGWO) 对其求解. 首先, 通过个体与决策个体的离散交叉操作, 实现算法在离散域中的更新, 提高了算法的运行效率. 然后, 引入了带评分策略的邻域搜索, 与单一的邻域操作相比, 评分策略能积累优质解信息并引导邻域搜索尽快到达优质解区域. 最后, 引入移除-插入机制对路径进行优化, 提高了解的多样性, 从而更好地进行全局寻优. 实验结果表明, 本文所提出的 DGWO 算法能有效缩短物流配送路径长度, 节约配送成本, 为求解 VRPSPDTW 问题提供有效的解决方案.

1 问题描述及数学模型

1.1 问题描述

VRPSPDTW 问题如图 1 所示, 其中正方形表示配送中心, 带编号的圆点表示客户编号, 箭头表示车辆配送路径. 若干相同类型的车辆从配送中心出发, 对一系列客户进行送货、取货服务, 在服务完所有客户后返回配送中心. 要求配送中心规划出一条合理的路线, 以最小的成本 (如车辆数目、最短行驶距等) 满足所有客户的需求, 并满足以下假设.

(1) 车辆到达客户点后, 需要先对客户送货再取货.

(2) 每个客户的需求量和接受服务的时间窗为已知的.

(3) 每辆车的载重量不能超过车辆最大容量.

(4) 每个客户只能被一辆车服务, 且客户的需求不能被拆分完成.

(5) 每个客户只在时间窗内接受服务.

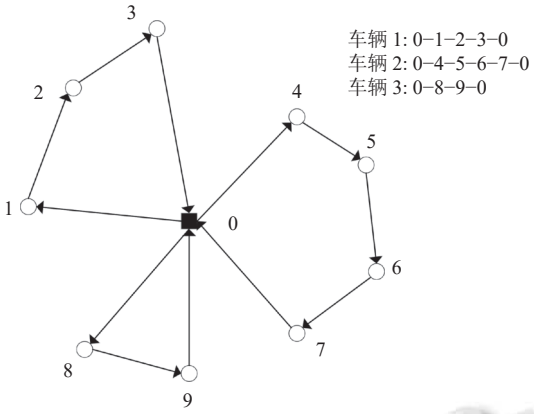

图 1 VRPSPDTW 图例

1.2 参数定义

为了能准确地描述模型, 本文所设计的参数和变量符号如表 1 所示.

表 1 符号表

类型	符号	说明
参数	Q	车辆最大载重
	D_i	客户i处的送货量
	P_i	客户i处的配送量
	s_i	客户i处的服务时间
	m	总的车辆数目, k表示车辆编号
	n	总的客户数目, i、j表示客户编号
	v	车辆行驶速度
	$[ET_i, LT_i]$	客户i接受服务的时间窗
	d_{ij}	客户i与j的距离
	T_i	车辆到达客户i的时间
	$propsize$	种群数量
	$Maxit$	算法最大迭代次数
	t	算法当前迭代次数
决策变量	y_{ijk}	由客户i到客户j的车辆k, 离开客户i的车辆载重
	x_{ijk}	决策变量, 若车辆k从客户i行驶到客户j, 则为1, 否则为0

1.3 数学模型

根据问题描述和参数定义, 建立如下的 VRPSPDTW 模型.

$$\min Z = \sum_{k=1}^{m}\sum_{j=0}^{n}\sum_{i=0}^{n} x_{ijk} d_{ij} \tag{1}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{n} x_{0jk} \leqslant m \tag{2}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{n} x_{ijk} = 1,\ j = 1,2,\cdots,n \tag{3}$$

$$\sum_{j=1}^{n} x_{0jk} = \sum_{i=1}^{n} x_{i0k},\ k = 1,2,\cdots,m \tag{4}$$

$$y_{ijk} \leqslant x_{ijk} Q,\ i,j = 1,2,\cdots,n;\ k = 1,2,\cdots,m \tag{5}$$

$$\sum_{j=1}^{n} y_{0jk} = \sum_{i=1}^{n} D_j \sum_{i=0}^{n} x_{ijk} \leqslant Q,\ k = 1,2,\cdots,m \tag{6}$$

$$\sum_{j=1}^{n} y_{i0k} = \sum_{i=1}^{n} P_i \sum_{i=0}^{n} x_{ijk} \leqslant Q,\ k = 1,2,\cdots,m \tag{7}$$

$$T_j = T_i + s_i + \frac{d_{ij}}{v},\ i,j = 1,2,\cdots,n \tag{8}$$

$$ET_j \leqslant T_j \leqslant LT_j,\ j = 1,2,\cdots,n \tag{9}$$

$$y_{ijk} \geqslant 0,\ i,j = 1,2,\cdots,n;\ k = 1,2,\cdots,m \tag{10}$$

$$x_{ijk} = \{0,1\},\ i,j = 1,2,\cdots,n,\ k = 1,2,\cdots,m \tag{11}$$

其中, 式 (1) 为目标函数, 即车辆行驶距离最小; 式 (2) 表示配送中心派出的车辆数目不超过最大车辆数目; 式 (3) 表示每个客户点只接受一辆车的服务; 式 (4) 表示每辆车需要从配送中心开始服务, 在完成配送任务后返回配送中心; 式 (5) 表示配送车辆在任意时刻的装载量均满足容量限制; 式 (6) 表示当车辆离开配送中心时, 负载量应满足所有配送客户的需求; 式 (7) 表示车辆返回配送中心时, 载重量为所服务的客户总的取货量; 式 (8) 表示车辆从客户i出发到达j的时间; 式 (9) 表示客户只在时间窗内接受服务; 式 (10) 和式 (11) 为变量约束.

由建立的模型可知, 相对于传统的 VRP 问题, VRPSPDTW 将时间窗和同时送取货考虑在内, 使模型的约束得到增加, 复杂了问题的求解空间. 随着客户数量的增加, VRPSPDTW 的求解空间和时间也呈指数增长.

1.4 构建带惩罚系数的适应度函数

对种群解码所得到的路径方案不一定都是可行的. 针对 VRPSPDTW 中存在的容量约束和时间窗约束, 将 2 种约束转化为具有惩罚值的软约束项. 在算法的更新过程中允许出现非可行解, 可采用适应度函数$F(x)$来评价解的优劣.

$$F(x) = f(x) + aq(x) + bt(x) \tag{12}$$

其中, $f(x)$为初始解x的目标函数值, 即车辆总行驶距离; $q(x)$为解x违反容量限制的总容量; $t(x)$为解x违反时间窗限制的总时间; a 和 b 分别为惩罚系数, 可设置较大

值. 若解x为可行解, 则$q(x)=t(x)=0$.

2 GWO 的基本原理

GWO 的优化过程包括等级分层、跟踪、包围和攻击猎物等步骤, 具体步骤如下.

等级分层: 计算种群的适应度, 将适应值最好的 3 匹狼设置为α、β和δ.

$$\vec{X}(t+1)=\vec{X}_p(t)-\vec{A}\cdot\left|\vec{C}\cdot\vec{X}_p(t)-\vec{X}(t)\right| \tag{13}$$

$$\vec{A}=2\vec{a}\cdot r_1-\vec{a} \tag{14}$$

$$\vec{C}=2r_2 \tag{15}$$

其中, t为当前迭代次数; $\vec{A}$和$\vec{C}$为系数向量; $\vec{X}(t)$为当前灰狼的位置向量, $\vec{X}_p(t)$为当前猎物的位置向量; r_1和r_2是 [0, 1] 的随机数; $\vec{a}$为收敛因子, 由 2 线性降到 0.

猎物的搜索过程主要是依靠α、β和δ引导完成, 其数学模型如下:

$$\begin{cases}\vec{D}_\alpha=\left|\vec{C}_1\cdot\vec{X}_\alpha-\vec{X}\right|\\ \vec{D}_\beta=\left|\vec{C}_2\cdot\vec{X}_\beta-\vec{X}\right|\\ \vec{D}_\delta=\left|\vec{C}_3\cdot\vec{X}_\delta-\vec{X}\right|\end{cases} \tag{16}$$

$$\begin{cases}\vec{X}_1=\left|\vec{X}_\alpha-\vec{A}_1\cdot\vec{D}_\alpha\right|\\ \vec{X}_2=\left|\vec{X}_\beta-\vec{A}_2\cdot\vec{D}_\beta\right|\\ \vec{X}_3=\left|\vec{X}_\delta-\vec{A}_3\cdot\vec{D}_\delta\right|\end{cases} \tag{17}$$

$$\vec{X}(t+1)=\frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3} \tag{18}$$

其中, $\vec{D}_\alpha$、$\vec{D}_\beta$和$\vec{D}_\delta$为α、β和δ与当前候选狼的距离; $\vec{X}_\alpha$、$\vec{X}_\beta$和$\vec{X}_\delta$是α、β和δ的位置向量; $\vec{C}_1$、$\vec{C}_2$和$\vec{C}_3$为系数向量; $\vec{A}_1$、$\vec{A}_2$和$\vec{A}_3$为系数向量.

3 离散灰狼算法

本文提出了一种离散灰狼优化算法 (DGWO) 解决上述 VRPSPDTW 问题. DGWO 的算法流程如图 2 所示, 在 DGWO 的初始化阶段, 为了提高初始种群的质量和算法的收敛性能, 提出了两种规则用于生成初始解. 在保留灰狼算法搜索机制的基础上, 设计了离散域的灰狼位置更新策略. 在局部搜索阶段, 采用 6 种有效的邻域操作, 并设计了改进的评分机制评价不同邻域操作的性能, 增强算法的局部寻优能力. 在全局搜索阶段, 将最不合理的路径剔除, 对优质解区域进行深入搜索. 本文后续将对 DGWO 各环节依次介绍.

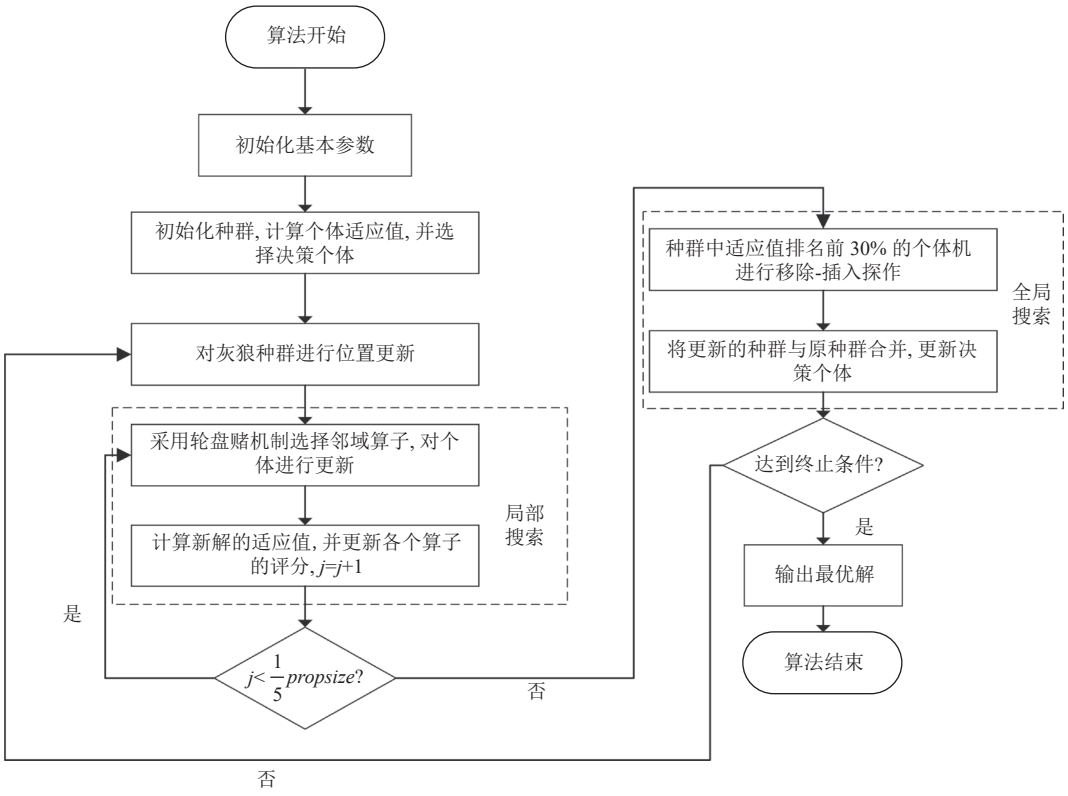

图 2 DGWO 算法流程图

3.1 编码与解码

本文采用整数编码, 若总的客户数目为n, 总的车辆数目为m, 则每个解x对应的长度为$n+m-1$. 0 表示配送中心, 大于 0 的整数表示客户点编号. 每辆车从配送中心 0 出发, 在完成一定数量的客户后返回配送中心. 假设客户点的集合为[1,2,3,4,5,6,7,8,9], 车辆编号的集合为[10,11,12]. 1 个可行的车辆路径编码和解码如图 3 所示.

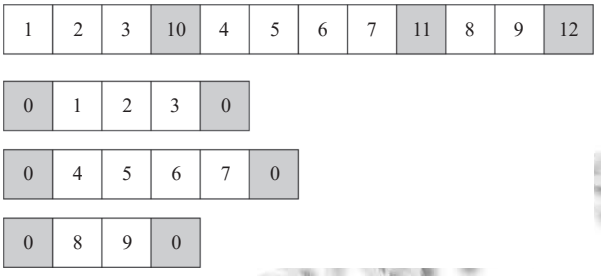

图 3 解的编码与解码

路径编码: 以车辆编号为结尾并包含客户点的一组数据序列; 路径解码: 车辆 10 依次服务客户 1、2、3; 车辆 11 依次服务客户 4、5、6、7; 车辆 12 依次服务客户 8、9.

采用以上的编码和解码方式能在一定程度上降低模型的复杂程度, 提高算法的运行效率.

3.2 种群初始化

初始化解的好坏直接影响算法的全局收敛性能和解的质量. 为了保证解的多样性和分散性, 本文采用 2 种规则构造初始解, 使用距离最短规则生成 1 个个体, 其余个体采用随机规则生成.

距离最短规则: 车辆从配送中心出发, 选择距离最短的客户i进行服务, 若存在多个距离最短客户, 则随机选择一个进行服务. 当车辆的剩余载量无法满足客户需求时, 车辆将返回配送中心. 配送中心将派出新车辆继续按上述方法对客户服务, 直至服务完所有的客户点.

随机规则: 采用随机的方式生成初始解, 随机规则应遵循两项原则. 1) 不允许产生相同的种群; 2) 随机产生的个体与已产生个体的适应度函数值应不小于数值τ.

在种群初始化中允许初始种群违反时间窗约束和容量约束, 即允许出现非可行解. 在算法的搜索过程中加入非可行解, 以增大算法的搜索范围, 避免算法陷入局部最优, 使算法在迭代过程中找到更优解.

3.3 离散灰狼优化机制

在基本的灰狼算法中, 根据式 (16)–式 (18) 来更新灰狼的位置, 参数A和C能平衡种群的勘探和开发能力. 参数A的递减可以保证了算法前期有较强的勘探能力, 后期有较强的开发能力. 参数C的随机性保证了种群多样性.

标准的 GWO 的参数和位置更新公式无法直接应用于离散优化问题, 因此对 GWO 中的参数A、C和位置更新策略进行改造, 改造的结果分别如下:

$$SN = S \times \left(1 - abs\left(\frac{A}{2}\right)\right) \tag{19}$$

$$OS = round(r_3(S - SN)) \tag{20}$$

$$OD = round\left(\frac{C}{2}(SN - OS)\right) \tag{21}$$

其中, r_3为 [0, 1] 的随机数, S为个体位置向量的长度; SN为个体位置向量中连续分量的长度; OS为连续分量的起始序列; OD为插入分量X_s的起始序列. 假设α的位置向量为X_α, 群狼ω的位置向量为X_ω. 如果X_ω的连续分量与X_α连续分量一致, 即表明灰狼个体不断接近领头狼的位置. 通过控制SN、OS和OD, 使狼群在一定范围内向领头狼的位置随机移动.

假设X_i^t为当前灰狼个体的位置, X_α^t, X_β^t和X_δ^t为当前领头狼的位置, 则离散灰狼算法的搜索机制如式 (22) 所示:

$$X_i^{t+1} = \begin{cases} LOX(X_i^t, X_\alpha, SN, OS, OD), & rand < \dfrac{1}{3} \\ LOX(X_i^t, X_\beta, SN, OS, OD), & \dfrac{1}{3} \leqslant rand \leqslant \dfrac{2}{3} \\ LOX(X_i^t, X_\delta, SN, OS, OD), & rand > \dfrac{2}{3} \end{cases} \tag{22}$$

其中, $rand$为[0,1]的随机数. 首先根据灰狼种群的适应度函数值, 选择出领头狼α、β和δ. 根据不同的 $rand$ 值, 将灰狼个体X_i^t与领头狼进行 LOX[20] 交叉操作. 图 4 给出了X_i^t与X_α^t进行 LOX 交叉操作的例子. 假设$SN = 4$, $OS = 4$, $OD = 3$, 将X_α^t中位置 4 到位置 8 的连续分量复制到X_s中, 可得到$X_s = [5,9,4,7]$, 并将X_i^t中包含X_s分量的部分删除. 根据 OD, 可确定X_s的插入顺序, 即从X_s的第 3 个分量开始依次插入到X_i^t的空白位置, 得到X_i^{t+1}, X_i^{t+1}为第i个种群个体的最新位置.

在求解组合优化问题时, DGWO 在离散域直接对灰狼位置向量进行更新, 不仅能保证解的收敛性和多样性, 并且能减少算法的额外计算量.

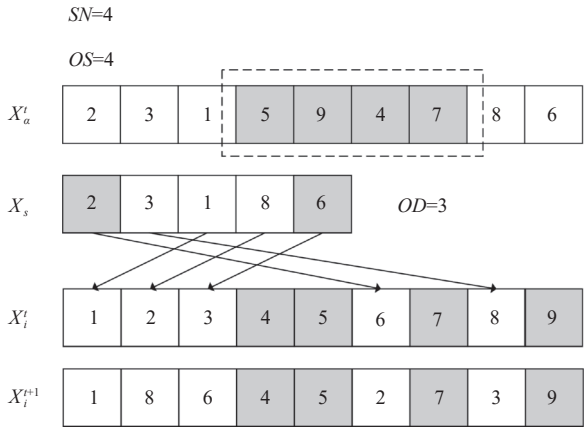

图 4 *LOX* 交叉

3.4 局部搜索

采用单一的邻域操作容易陷入该结构的局部最优. 故采用多种邻域操作相结合的邻域结构进行探索, 从而更容易获得优质解, 减少搜索的盲目性. 针对 VRPSPDTW 的具体问题, 本文采用了路径内与路径间相结合的邻域操作进行局部搜索. 根据算子在算法中的表现, 不断调整每个算子的权重, 提高优异算子的选中概率. 探索能力较强的算子将会有更高的概率被选择, 而其他算子也有一定的概率被选择, 这能形成众多新的邻域组合结构. 下面对各种邻域操作进行介绍.

路径内:

(1) 2-opt 算子: 先随机选取一条车辆路径 r, 在 r 中任意选取两个客户点, 将两个客户间的顺序进行逆转. 如图 5 所示, 选择 0-1-2-3-4-0 路径中的客户点 1 和客户点 4, 进行 2-opt 操作, 得到新路径 0-1-3-2-4-0.

图 5 2-opt 算子

(2) exchange 算子: 随机选取一条车辆路径 r, 在路径 r 任意选取两个客户点进行交换, 形成一条新路径.

(3) relocate 算子: 随机选取一条路径 r, 在路径 r 中随机选取 m 个连续客户点, 将这 m 个客户点重新定位到该路径中.

路径间:

(4) shift 算子: 首先随机选取两条路径 $r1$ 和 $r2$, 并从 $r1$ 中随机选择一个客户点i, 将i插入 $r2$ 中以生成两条新路径. 如图 6 所示, 假设当前存在两条路径, 分别为 0-1-2-3-4-0 和 0-5-6-7-8-0. 将客户点 1 从原路径中剔除, 并插入到另外一条路径中, 得到两条新路径, 分别为 0-2-3-4-0 和 0-1-5-6-7-8-0.

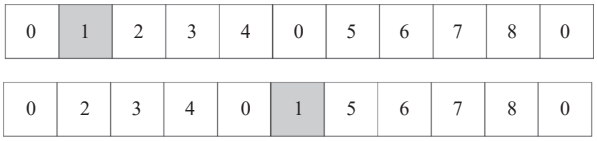

图 6 shift 算子

(5) swap 算子: 随机选取 2 条路径 $r1$ 和 $r2$, 分别在 2 条路径中选取一个客户点, 交换 2 个客户点的位置.

(6) gene 算子: 随机选取 2 条路径 $r1$ 和 $r2$, 并随机选取一条路径中的一个客户点, 插入到另一条路径中, 插入位置为距离该客户点最近的 2 个节点之间.

在执行完任意一种局部操作后, 若产生的解为可行解且优于原解, 则采用新解代替原解, 否则, 不接受该解.

3.5 自适应选择邻域策略

采用不同形式的邻域操作, 有助于避免陷入局部最优并提高解的搜索质量. 在算法的每次迭代中, DGWO 采用轮盘赌的机制选择多种邻域结构来实现路径的更新. 根据更新后路径的优劣, 对每个邻域结构进行评分. 假设每个算子的初始得分为P_s, 最高得分为$P_{\max}$, 最低得分为$P_{\min}$, $F(x)$为解x的适应值, 则邻域操作的评分机制根据式 (23) 进行更新:

$$P_i^j = P_i^{j-1} + \sigma \frac{F(x)}{F(x')} - 1 \tag{23}$$

其中, P_i^j为第i个邻域结构第j次使用后的得分; x为局部操作前的个体; x'为局部操作后的个体. σ存在 3 种情况, 若局部操作后的解优于决策个体, 即α、β和δ, 则σ为σ_1; 若局部操作后的解优于原解, 则σ为σ_2; 若局部操作后的解比当前解差, 则σ为σ_3. σ_1、σ_2和σ_3均为常数, 在本文中取$\sigma_1 = 1.5$, $\sigma_2 = 1$, $\sigma_3 = 0.8$. 当局部操作被执行了M次后, 算子j被选中的概率为$\dfrac{P_j^M}{\sum_{j=1}^{h} P_j^M}$, 其中$h$为算子的总个数. 若$P_i^j$大于$P_{\max}$或$P_i^j$小于$P_{\min}$, 则按照式 (24) 进行取值:

$$P_i^j = \begin{cases} P_{\max}, & P_i^j > P_{\max} \\ P_{\min}, & P_i^j < P_{\min} \end{cases} \tag{24}$$

为了能在较短的时间内平衡算法搜索的广度和深

度, 选择适应值前 10% 的种群和适应值后 10% 的种群进行局部搜索.

3.6 移除-插入操作

路径寻优过程中, 采用移除算子和插入算子全局优化路径. 移除算子将部分客户点从路径中移除, 再通过插入算子, 将移除的客户点插入到合理的路径中, 得到新路径. 如图 7 所示, 黑色圆点为移除的客户点, 将客户点重新插入路径中, 形成了新路径, 新路径与原路径相比减少了总距离. 下面对本文采用的移除-插入机制进行介绍.

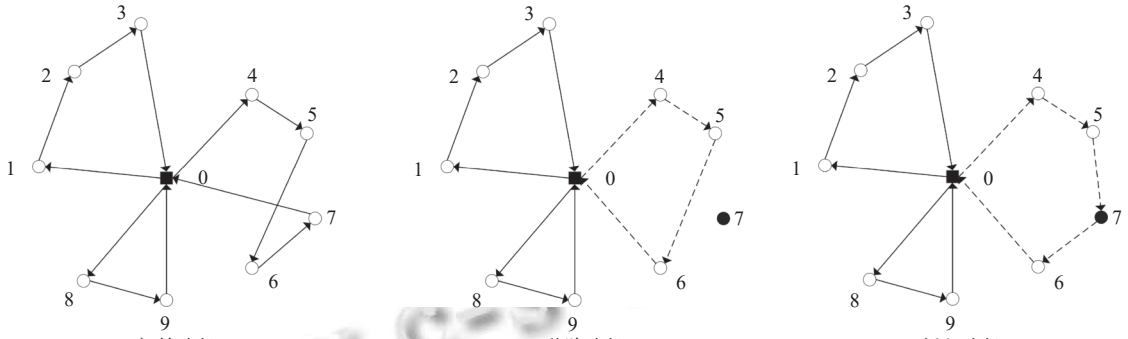

图 7 移除-插入操作

(1) 选择最不经济的一条路线, 即平均行驶距离最大的一条路线. 将最不经济的一条路线移除该路线上的所有客户点为待插入客户点. 平均行驶距离的计算公式如下:

$$A_k = \frac{D(k)}{N_k} \tag{25}$$

其中, $D(k)$表示第k辆车的总行驶距离, N_k为第k辆车所服务的客户数量.

(2) 令所有待插入客户点的集合为 R, 在满足时间窗和容量约束的基础上, 记录每个插入点的距离增量. 若将客户点k插入到客户点i、j中, 则距离增量为 $\Delta d(i,k,j) = d_{ik} + d_{kj} - d_{ij}$.

(3) 选择距离增量小的位置插入, 并将该客户点从 R 中移除, 重复操作直到 R 为空集.

与随机移除算子相比, 本文选用的算子考虑了路线的整体性. 随机移除算子随机性强, 跳出局部搜索能力较强, 但容易将正确位置的客户点移除, 产生无效操作. 若选择最不经济的一条路线进行移除, 则会集中移除一条路线中的客户点, 策略简单, 且不易陷入局部最优.

3.7 迭代终止条件

算法存在两种终止条件. 1) 算法设置最大迭代次数, 若当前算法的迭代次数大于算法最大迭代次数, 则算法终止; 2) 令$F(x)$为当前最优值, 若$F(x)$的数值连续ρ次保持不变, 则算法终止.

4 实验与分析

实验运行环境如下: 计算机为 Lenovo XiaoXin Air 15ARE 2021, 处理器为 AMD Ryzen 7 4800U with Radeon Graphics, 主频为 1.80 GHz, 内存为 16.0 GB. 采用 Matlab R2020b 软件进行模拟仿真. 算法的参数设置如下: *Maxit*为 100, *propsize* 为 100, P_s为 50, $P_{\max}$为 200, $P_{\min}$为 1. 为说明 DGWO 的有效性和实用性, 选取 4 组实验进行对比. 实验 1 对算法变体进行实验比较, 实验 2 对小规模的标准算例进行实验, 实验 3 对大规模标准算例进行实验, 实验 4 对真实案例进行实验分析.

4.1 DGWO 变体实验分析

为了测试多规则初始化种群策略、自适应选择邻域策略和移除-插入机制的有效性, 在本实验中, 我们研究 DGWO 的相关变体.

(1) Variant-1: DGWO 采用随机策略构造初始解.

(2) Variant-2: DGWO 无自适应选择邻域策略.

(3) Variant-3: DGWO 无移除-插入机制.

以标准数据集中的 Rdp101 为例, 分析 DGWO 和 Variant-1 的收敛性能. DGWO 和 Variant-1 的最优结果迭代过程如图 8 所示. 以 Cdp2 问题为例, 表 2 对比了 DGWO 和 Variant 运行 20 次的最优值、平均值和最差值.

从图 8 中看出, DGWO 迭代到 75 次左右, 算法收

敛到最优解 1 646.20; Variant-1 迭代到 85 次左右, 算法收敛到最优解 1 649.57. 这说明采用多策略构造初始种群能提高初始解的质量并增强初始解的多样性, 加快种群的收敛并保持良好寻优能力.

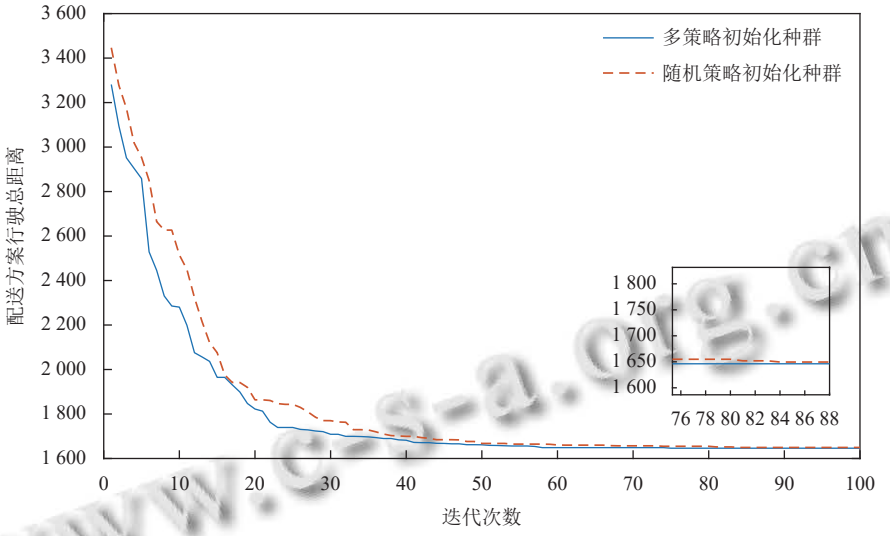

图 8 DGWO 和 Variant-1 在 Rdp101 算例上的收敛曲线图

表 2 DGWO 和 Variant-2、Variant3 的比较结果

算例	DGWO			Variant-2			Variant-3		
	最优	平均	最差	最优	平均	最差	最优	平均	最差
Cdp201	**591.6**	**591.56**	**591.56**	591.56	593.12	594.67	605.53	607.98	610.43
Cdp202	**591.6**	**591.56**	**591.56**	593.43	596.08	598.73	602.30	608.86	615.42
Cdp203	**591.7**	**591.17**	**591.17**	592.35	594.44	596.52	604.34	612.50	620.65
Cdp204	**591.7**	**591.17**	**591.17**	595.30	596.38	597.45	601.75	608.20	614.64
Cdp205	**588.8**	**588.88**	**588.88**	588.88	593.60	598.32	594.21	606.82	619.43
Cdp206	**588.9**	**588.49**	**588.49**	589.36	591.57	593.78	595.01	606.73	618.45
Cdp207	**588.9**	**588.29**	**588.29**	590.39	593.42	596.45	598.32	610.71	623.10
Cdp208	**588.2**	**588.32**	**588.32**	589.62	590.41	591.20	593.50	605.70	617.89

从表 2 中可以看出, DGWO 的最优值、平均值和最差值均优于 Variant-2 和 Variant-3, 这表明 DGWO 拥有更好的稳定和收敛性能. 以 Cdp202 为例, DGWO 最优值相比 Variant-2 提高了 0.3%, 这表明引入自适应选择邻域策略, 动态构造了丰富的邻域结构, 能增强算法在优质解区域的探索深度.

以 Cdp202 为例, DGWO 最优值相比 Variant-3 提高了 1.77%, 这表明移除-插入机制能有效地跳出局部最优, 提升算法的综合性能.

因此 3 种策略对于算法的优化存在积极作用.

4.2 小规模算例实验分析

选用 Wang 等[10] 设计的 9 种中小规模算例进行测试, 包括 3 组 10 个客户点规模, 3 组 25 个客户点规模和 3 组 50 个客户点规模的算例. 表 3 列出了 DGWO、p-SA[21]、IGAFSA[11] 和 DCS[12] 在小规模算例的最优解对比结果.

表 3 小规模算例实验结果对比

算例/客户数	p-SA	IGAFSA	DCS	DGWO
Rdp1001/10	349.98	348.98	349.98	**348.98**
Rdp1004/10	216.69	216.69	216.69	**216.69**
Rdp1007/10	310.81	310.81	310.81	**310.81**
RCdp2501/25	551.05	552.21	551.05	**551.05**
RCdp2504/25	473.46	472.23	473.46	**472.23**
RCdp2507/25	540.87	540.87	540.87	**540.87**
RCdp5001/50	994.18	994.80	994.18	**994.18**
RCdp2005/50	725.59	723.58	725.59	**723.58**
RCdp5007/50	809.72	808.54	809.72	**808.54**

表 3 的实验结果表明, 在客户点规模较小的时候, DGWO、p-SA、IGAFSA 和 DCS 都能得到问题的最优解. 当数据规模变大时, 数据的计算复杂程度会显著

增加. 当数据规模扩展到 25 时, 只有 DGWO 均能得到问题的最优解. 当数据规模扩展到 50 时, 只有 DGWO 和 IGAFSA 能每次都准确得到问题的最优解. 这验证了 DGWO 在处理小规模算例的有效性.

4.3 大规模算例实验分析

选取 Wang 等[10] 针对 VRPSPDTW 问题设计的测试集作为验证算例. 共 56 个数据集, 由 6 种不同的问题类型组成, 分别为 Rdp1、Rdp2、Cdp1、Cdp2、RCdp1 和 RCdp2, 每个算例都包含 100 个客户点. 其中 Rdp1 和 Rdp2 的客户点为随机分配, Cdp1 和 Cdp2 的客户点满足聚类分布, Rcdp1 和 Rcdp2 的客户点混合了随机和聚类分布. 记录 DGWO 算法 20 次独立运行中的最优结果. 将实验结果与 p-SA 算法[11]、DCS 算法[12]、VNS-BSTS 算法[21]、SA-ALNS 算法[22] 进行对比. DGWO 与上述算法在 Cdp, Rdp, RCdp 算例的对比结果分别如表 4–表 6 所示, 其中粗体表示更优值.

表 4 Cdp 算例实验结果对比

算例	p-SA	DCS	VNS-BSTS	SA-ALNS	DGWO
Cdp101	992.88	998.29	976.04	**936.96**	967.42
Cdp102	955.31	954.31	942.45	**935.50**	936.74
Cdp103	958.66	923.05	896.28	**888.52**	890.23
Cdp104	944.74	931.26	872.39	**881.94**	885.79
Cdp105	989.86	981.45	1 080.63	983.10	**981.45**
Cdp106	878.29	878.45	963.45	878.29	**878.29**
Cdp107	911.90	912.37	987.64	**908.99**	915.64
Cdp108	1 063.73	978.82	934.41	**936.10**	924.65
Cdp109	947.90	940.49	**909.27**	932.69	922.67
Cdp201	591.56	591.56	591.56	591.56	**591.56**
Cdp202	591.56	591.56	591.56	591.56	**591.56**
Cdp203	591.17	591.17	591.17	591.17	**591.17**
Cdp204	590.60	590.60	599.33	591.17	**591.17**
Cdp205	588.88	588.88	588.88	588.88	**588.88**
Cdp206	588.49	588.49	588.49	588.49	**588.49**
Cdp207	588.29	588.29	588.29	588.29	**588.29**
Cdp208	588.32	588.32	588.32	588.32	**588.32**

(1) 从表 4 可以看出, 在 Cdp105、Cdp106 算例中, DGWO 的实验结果最优; 在 Cdp109 算例中, VNS-BSTS 的实验结果最优; 在其余 Cdp1 算例中, SA-ALNS 均能获得最优解. 总的来说, DGWO 在处理 Cdp1 算例上的求解性能弱于 SA-ALNS. 在 Cdp2 算例中, DGWO 均能获得最优解. 从表 5 可以看出, 在 RCdp103、RCdp201、RCdp202、RCdp203、RCdp205、RCdp206 和 RCdp207 上, DGWO 均刷新了当前已知最优解. 在其余 RCdp 问题上, DGWO 的优化性能略弱于 SA-ALNS 和 VNS-BSTS. 从表 6 可以看出, 除了 Rdp103、Rdp104、Rdp105、Rdp110、Rdp111 和 Rdp112, DGWO 在其余 Rdp 算例中均更新了当前已知最优解.

表 5 Rdp 算例实验结果对比

算例	p-SA	DCS	VNS-BSTS	SA-ALNS	DGWO
Rdp101	1 660.98	1 658.65	1 650.80	1 650.20	**1 646.27**
Rdp102	1 491.75	1 490.13	1 486.12	1 485.47	**1 477.60**
Rdp103	**1 226.77**	1 228.48	1 294.75	1 228.10	1 234.60
Rdp104	1 000.65	1 005.99	**984.81**	1 011.50	1 012.03
Rdp105	1 399.81	**1 340.06**	1 377.11	1 366.87	1 345.76
Rdp106	1 275.69	1 270.29	1 261.50	1 260.90	**1 256.76**
Rdp107	1 082.92	1 084.00	1 144.02	1 085.40	**1 076.49**
Rdp108	962.48	964.38	968.32	967.40	**959.90**
Rdp109	1 181.92	1 156.90	1 224.86	1 155.75	**1 151.96**
Rdp110	1 106.52	1 108.81	**1 101.33**	1 095.23	1 130.63
Rdp111	1 073.62	1 077.65	1 117.76	**1 069.50**	1 084.35
Rdp112	966.06	977.59	**961.29**	963.77	962.36
Rdp201	1 286.55	12 181.63	1 254.57	1 181.73	**1 177.92**
Rdp202	1 150.31	1 152.65	1 202.27	1 065.16	**1 039.02**
Rdp203	997.84	950.79	949.42	927.65	**885.70**
Rdp204	848.01	776.00	837.13	790.96	**748.13**
Rdp205	1 046.06	1 051.38	1 027.49	1 000.14	**986.12**
Rdp206	959.94	957.81	938.63	948.28	**894.48**
Rdp207	899.82	890.52	912.26	849.44	**809.41**
Rdp208	739.06	737.05	737.26	728.83	**719.60**
Rdp209	947.80	930.26	940.29	909.35	**871.14**
Rdp210	1 005.11	1 005.11	945.97	939.80	**921.91**
Rdp211	812.44	819.88	805.22	774.99	**772.36**

表 6 RCdp 算例实验结果对比

算例	p-SA	DCS	VNS-BSTS	SA-ALNS	DGWO
RCdp101	1 659.59	**1 654.32**	1 708.21	1 666.12	1 664.79
RCdp102	1 522.76	1 522.76	1 526.36	**1 494.01**	1 500.12
RCdp103	1 344.62	1 344.63	1 336.05	1 351.66	**1 334.65**
RCdp104	1 268.43	1 269.31	**1 177.21**	1 182.42	1 226.51
RCdp105	1 581.54	1 581.26	**1 548.38**	1 566.96	1 557.46
RCdp106	1 418.16	1 419.26	**1 408.19**	1 424.07	1 420.46
RCdp107	1 360.17	1 360.17	**1 295.43**	1 295.88	1 304.31
RCdp108	1 169.57	1 170.12	1 207.60	**1 164.45**	1 167.82
RCdp201	1 513.72	1 520.56	1 437.48	1 326.19	**1 304.13**
RCdp202	1 273.26	1 242.92	1 412.52	1 159.19	**1 114.42**
RCdp203	1 123.58	1 087.37	1 064.95	959.17	**957.63**
RCdp204	897.14	822.02	813.74	**796.12**	808.50
RCdp205	1 357.44	1 357.44	1 316.06	1 188.50	**1 164.32**
RCdp206	1 166.88	1 166.88	1 154.36	1 139.60	**1 076.57**
RCdp207	1 089.85	1 093.37	1 098.64	1 023.51	**966.37**
RCdp208	862.89	862.89	843.30	**793.98**	796.04

(2) 在 56 个算例中, DGWO 在 85.7% 的算例上优于 p-SA, 在 83.9% 的算例上优于 DCS, 在 83.9% 的算例上优于 VNS-BSTS, 在 73.2% 的算例上优于 SA-ALNS. 这表明本算法在求解 VRPSPDTW 问题上具有较好的性能, 整体性能优于 p-SA、DCS、VNS-BSTS 和

SA-ALNS.

(3) 对比实验表明 DGWO 算法在满足随机分布的算例中表现优异, 输出结果稳定. 图 9 分别给出了算例 Rdp、RCdp 和 Cdp 的部分路径图. 图 10 给出了 Rdp108 算例的收敛曲线, 这表明 DGWO 能以较快的收敛速度寻找到更优解.

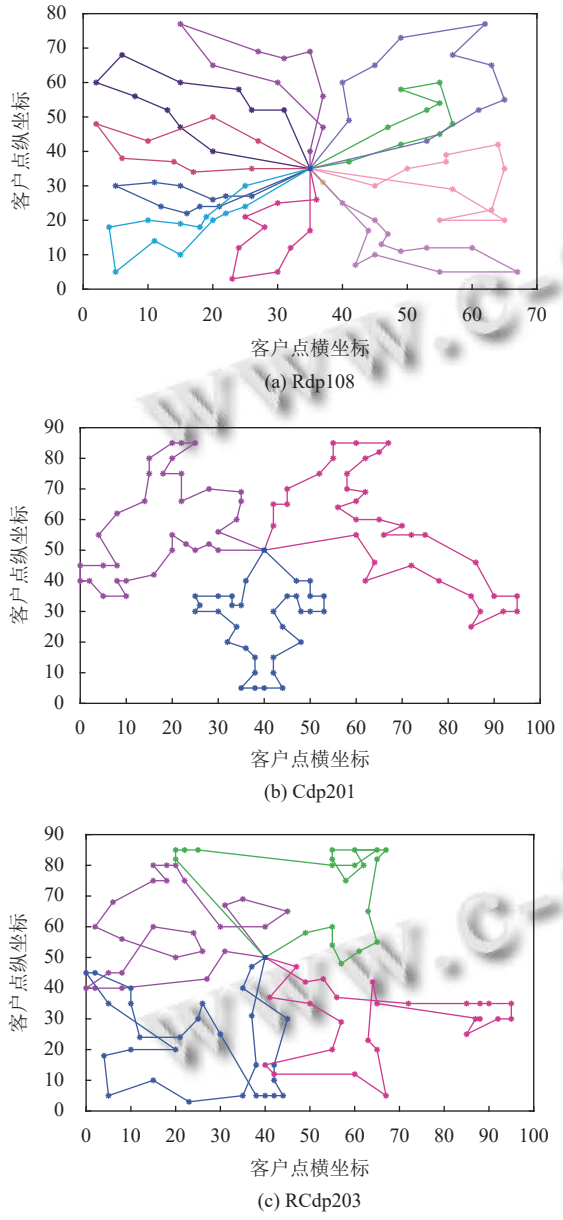

图 9 部分算例最优路线

综上表明, 本文提出的 DGWO 算法具有较好的收敛性和可行性, 在求 VRPSPDTW 问题中具有良好的效果同时也说明不同的机制组合很大程度上法的性能.

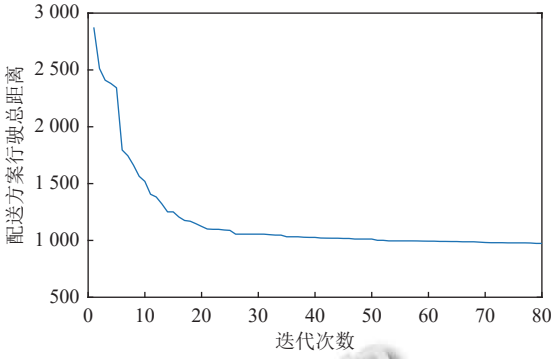

图 10 Rdp108 收敛曲线

4.4 应用案例分析

为了验证本算法的实际有效性, 本文选择南京市某一连锁超市的部分数据, 包括 1 个配送中心和 18 个需要配送的连锁店. 参与配送的车辆共 4 台, 且车辆类型统一. 放了方便研究, 本文对 18 个连锁店进行了 1, 2, …, 18 编号处理, 数字 0 表示配送中心. 各连锁店的具体数据如表 7 所示. 配送车辆的最大载重量为 3 吨, 在不考虑交通拥堵的情况下, 假设车辆平均行驶速度为 50 km/h. 本文的主要目标是最小化总行驶距离. 图 11 为车辆最优配送路线, 图 12 为算法的收敛曲线, 可以看出 DGWO 能在较短的时间内获得最优解, 这表明 DGWO 能应用于实际问题.

表 7 各连锁店的数据信息

编号	坐标 (经度, 纬度)	送货量 (吨)	取货量 (吨)	时间窗	服务时间 (min)
0	118.84220, 32.13516	0	0	5:30–17:00	0
1	118.67389, 32.18331	0.7	0.2	6:00–10:00	18
2	118.71460, 32.15154	0.3	0	6:30–8:00	6
3	118.73654, 32.14519	0.3	0	6:30–9:40	6
4	118.73798, 32.12367	0.4	0.1	7:00–10:30	10
5	118.71257, 32.21605	0.7	0	7:00–9:30	14
6	118.73682, 32.21092	0.8	0.2	7:20–10:20	20
7	118.75963, 32.23461	0.4	0.1	6:20–9:30	10
8	118.76829, 32.23851	0.4	0.1	6:30–10:00	10
9	118.75588, 32.24828	0.6	0.1	7:00–9:30	14
10	118.77869, 32.10558	0.7	0.2	7:30–10:00	18
11	118.78619, 32.10729	0.6	0.1	6:20–9:00	14
12	118.80871, 32.11022	0.6	0.2	6:30–9:30	16
13	118.83152, 32.11927	0.6	0.1	6:30–8:00	14
14	118.84256, 32.12427	0.8	0.2	6:30–9:20	20
15	118.85721, 32.11609	0.8	0.1	6:00–9:00	18
16	118.87165, 32.12710	0.9	0.3	7:00–10:00	24
17	118.87049, 32.12074	0.9	0.2	6:30–9:20	22
18	118.87338, 32.11585	0.9	0.2	7:00–9:30	22

5 结束语

本文针对带时间窗和同时取送货的车辆路径优化

问题, 建立了以行驶距离最小为目标的车辆优化模型, 设计离散灰狼优化算法 DGWO 进行求解. DGWO 具有以下优点: 1) 在种群初始化中允许加入非可行解, 增大算法的搜索范围; 2) 设计了在离散空间中灰狼的位置更新策略, 使 GWO 能处理离散优化问题; 3) 采用评分机制丰富了邻域操作, 拓展了搜索深度, 从而改善了算法的性能; 4) 采用移除-插入操作, 平衡了算法的搜索深度和广度, 加速的解的收敛. 通过仿真实验和对比, 验证了 DGWO 是求解 VRPSPDTW 问题的有效算法, 该算法能在较短时间内获得最优解. 后续将进一步研究基于模糊需求的车辆路径优化问题, 并设计有效算法来求解.

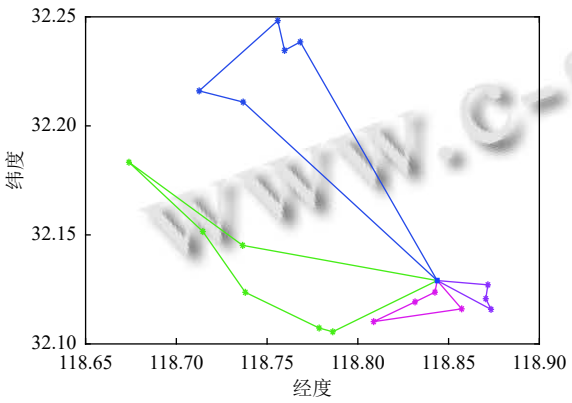

图 11 最优配送路线

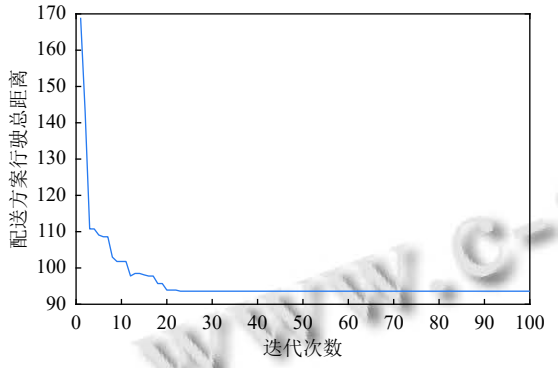

图 12 收敛曲线

参考文献

1 Zhang SQ, Mu D, Wang C. A solution for the full-load collection vehicle routing problem with multiple trips and demands: An application in Beijing. IEEE Access, 2020, 8: 89381–89394.

2 Pureza V, Morabito R, Reimann M. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. European Journal of Operational Research, 2012, 218(3): 636–647. [doi: 10.1016/j.ejor.2011.12.005]

3 Alyasiry AM, Forbes M, Bulmer M. An exact algorithm for the pickup and delivery problem with time windows and last-in-first-out loading. Transportation Science, 2019, 53(6): 1695–1705. [doi: 10.1287/trsc.2019.0905]

4 Angelelli E, Mansini R. The vehicle routing problem with time windows and simultaneous pick-up and delivery. In: Klose A, Speranza MG, Wassenhove LN, eds. Quantitative Approaches to Distribution Logistics and Supply Chain Management. Berlin Heidelberg: Springer , 2002. 249–267.

5 Bettinelli A, Cacchiani V, Crainic TG, *et al*. A branch-and-cut-and-price algorithm for the multi-trip separate pickup and delivery problem with time windows at customers and facilities. European Journal of Operational Research, 2019, 279(3): 824–839. [doi: 10.1016/j.ejor.2019.06.032]

6 Sitek P, Wikarek J, Rutczyńska-Wdowiak K, *et al*. Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. Neurocomputing, 2021, 423: 670–678. [doi: 10.1016/j.neucom.2020.02.126]

7 张九龙, 王晓峰, 芦磊, 等. 若干新型智能优化算法对比分析研究. 计算机科学与探索, 2022, 16(1): 88–105.

8 钟倩漪, 钱谦, 伏云发, 等. 粒子群优化算法在关联规则挖掘中的研究综述. 计算机科学与探索, 2021, 15(5): 777–793.

9 李丹. 改进群智能优化算法的海上物流配送路径优化方法. 舰船科学技术, 2020, 42(16): 184–186.

10 Wang HF, Chen YY. A genetic algorithm for the simultaneous delivery and pickup problems with time window. Computers & Industrial Engineering, 2012, 62(1): 84–95.

11 黄务兰, 张涛. 基于改进全局人工鱼群算法的 VRPSPDTW 研究. 计算机工程与应用, 2016, 52(21): 21–29.

12 王超, 刘超, 穆东, 等. 基于离散布谷鸟算法求解带时间窗和同时取送货的车辆路径问题. 计算机集成制造系统, 2018, 24(3): 570–582.

13 Belgin O, Karaoglan I, Altiparmak F. Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. Computers & Industrial Engineering, 2018, 115: 1–16.

14 Lagos C, Guerrero G, Cabrera E, *et al*. An improved particle swarm optimization algorithm for the VRP with simultaneous pickup and delivery and time windows. IEEE Latin America Transactions, 2018, 16(6): 1732–1740. [doi: 10.1109/TLA.2018.8444393]

15 Wang Y, Li Q, Guan XY, *et al*. Collaborative multi-depot

pickup and delivery vehicle routing problem with split loads and time windows. Knowledge-based Systems, 2021, 231: 107412. [doi: 10.1016/j.knosys.2021.107412]

16 Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software, 2014, 69: 46–61. [doi: 10.1016/j.advengsoft.2013.12.007]

17 黄戈文, 蔡延光, 戚远航, 等. 自适应遗传灰狼优化算法求解带容量约束的车辆路径问题. 电子学报, 2019, 47(12): 2602–2610.

18 姜天华. 混合灰狼优化算法求解柔性作业车间调度问题. 控制与决策, 2018, 33(3): 503–508.

19 顾九春, 姜天华, 朱惠琦. 多目标离散灰狼优化算法求解作业车间节能调度问题. 计算机集成制造系统, 2021, 27(8): 2295–2306.

20 Della Croce F, Tadei R, Volta G. A genetic algorithm for the job shop problem. Computers & Operations Research, 1995, 22(1): 15–24.

21 Ma YF, Li ZM, Yan F, *et al*. A hybrid priority-based genetic algorithm for simultaneous pickup and delivery problems in reverse logistics with time windows and multiple decision-makers. Soft Computing, 2019, 23(15): 6697–6714. [doi: 10.1007/s00500-019-03754-5]

22 Shi Y, Zhou YJ, Boudouh T, *et al*. A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup-delivery and time window. Engineering Applications of Artificial Intelligence, 2020, 95: 103901. [doi: 10.1016/j.engappai.2020.103901]

(校对责编: 牛欣悦)