

# 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

黄戈文<sup>1,3</sup>, 蔡延光<sup>1</sup>, 戚远航<sup>2</sup>, 陈厚仁<sup>1</sup>, 王世豪<sup>1</sup>

(1. 广东工业大学自动化学院, 广东广州 510006; 2. 电子科技大学中山学院计算机学院, 广东中山 528402;  
3. 嘉应学院信息网络中心, 广东梅州 514015)

**摘 要:** 带容量约束的车辆路径问题是 NP 难的组合优化问题, 精确算法无法在合理的时间内得到有效的解。本文提出了一种采用灰狼空间整数编码和先路由后分组解决方案生成策略的自适应遗传灰狼优化算法用于求解带容量约束的车辆路径问题。该算法提出了移动平均自适应灰狼更新策略和灰狼基因遗传策略提高全局收敛能力, 同时提出带 3-opt 的劣势点启发邻域搜索策略来增强算法的全局和局部搜索能力。实验结果表明: 所提出算法具有较高的计算精度和较强的寻优能力, 有较高的鲁棒性, 通过与自适应扫描和速度推测粒子群优化算法、K 均值聚类灰狼优化混合算法、大邻域搜索和蚁群优化混合算法、基于精英选择的多种群人工蜂群算法、基于集覆盖的扩展节省算法、混合变邻域生物共栖搜索算法等 6 个算法对比证明了算法的有效性。

**关键词:** 组合优化; 车辆路径问题; 离散灰狼优化算法; 自适应更新; 遗传操作; 邻域搜索

**中图分类号:** TP301 **文献标识码:** A **文章编号:** 0372-2112 (2019)12-2602-09

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2019.12.021

## Adaptive Genetic Grey Wolf Optimizer Algorithm for Capacitated Vehicle Routing Problem

HUANG Ge-wen<sup>1,3</sup>, CAI Yan-guang<sup>1</sup>, QI Yuan-hang<sup>2</sup>, CHEN Hou-ren<sup>1</sup>, WANG Shi-hao<sup>1</sup>

(1. School of Automation, Guangdong University of Technology, Guangzhou, Guangdong 510006, China;

2. School of Computer Science, University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan, Guangdong 528402, China;

3. Information and Network Center, Jiaying University, Meizhou, Guangdong 514015, China)

**Abstract:** Capacitated vehicle routing problem (CVRP) is an NP-hard combinatorial optimization problem. Many CVRP instances cannot be solved by the exact algorithms in a reasonable time. This paper presents an adaptive genetic grey wolf optimizer algorithm (AGGWOA), which implements grey wolf space integer coding and route-first cluster-second solution generation strategy, to solve the capacitated vehicle routing problem. The AGGWOA proposes the adaptive update strategy on moving average and grey wolf genetic operation that improve the global convergence of the algorithm. To enhance the global search ability and the local search ability of the algorithm, the AGGWOA proposes the inferior-node heuristic neighborhood search strategy, which implements the 3-opt local search operation. The experimental results indicate that the algorithm proposed has superior computational accuracy, effective optimization ability and high robustness. The effectiveness of the algorithm proposed is proved by comparing AGGWOA with 6 other algorithms including adaptive sweep plus velocity tentative PSO (Adaptive Sweep + VTPSO), K-means clustering GWO (K-GWO), hybrid large neighbourhood search algorithm with ant colony optimization (LNS-ACO), elitism-based multiple colonies artificial bee colony (EBMC-ABC), set-covering-based extended savings algorithm (SC-ESA), hybrid variable neighborhood symbiotic organisms search (HVNSOS).

**Key words:** combination optimization; vehicle routing; discrete grey wolf optimizer; adaptive update; genetic operation; neighborhood search

收稿日期: 2018-12-02; 修回日期: 2019-07-20; 责任编辑: 蓝红杰

基金项目: 国家自然科学基金 (No. 61074147); 广东省自然科学基金 (No. S2011010005059); 广东省教育部产学研结合项目 (No. 2012B091000171, No. 2011B090400460); 广东省科技计划项目 (No. 2012B050600028, No. 2014B010118004, No. 2015A030401104, No. 2016A050502060); 广东省普通高校青年创新人才项目 (No. 2018KQNCX333); 广州市花都区科技计划项目 (No. HD14ZD001); 广州市科技计划项目 (No. 201604016055); 广州市天河区科技计划项目 (No. 2018CX005)

## 1 引言

带容量约束的车辆路径问题 (Capacitated Vehicle Routing Problem, CVRP) 是一类典型的车辆路径问题 (Vehicle Routing Problem, VRP)<sup>[1]</sup>, 其特征是客户具有一定的货物运输需求, 车辆则具有最大容量限制. CVRP 是 NP-Hard 组合优化问题, 大规模的 CVRP 算例很难应用传统算法进行求解. 近年来很多学者应用群体智能算法求解 CVRP. Korayem 等<sup>[2]</sup>提出了先分组后路由的 K 均值聚类法和灰狼优化混合算法用于求解 CVRP 问题. Akpinar<sup>[3]</sup>提出了用于 CVRP 的大邻域搜索和蚁群优化混合算法. Ng 等<sup>[4]</sup>提出了用于带重新路由策略 CVRP 的多种群人工蜂群算法. Akhand 等<sup>[5]</sup>提出了自适应扫描和速度推测粒子群优化算法用于路径优化. Hannan 等<sup>[6]</sup>提出了改进粒子群优化算法实现垃圾收集最优路径求解. 李阳和范厚明<sup>[7]</sup>提出了混合变邻域生物共栖搜索算法用于求解 CVRP.

灰狼优化算法<sup>[8]</sup>是模仿自然界中灰狼群领导层级机制和狩猎过程的群智能优化方法. 算法具有全局搜索能力强, 收敛快, 控制参数少, 实现容易等优点, 已经在函数寻优、机器学习、数据挖掘、控制器设计调优、电力调度等方面得到有效应用<sup>[8,9]</sup>, 但目前很少学者将灰狼优化算法用于 VRP.

因此, 本文提出自适应遗传灰狼优化算法 (Adaptive Genetic Grey Wolf Optimizer Algorithm, AGGWOA) 求解 CVRP, 是在传统灰狼优化算法基础上采用灰狼空间整数编码和先路由后分组解决方案生成策略实现灰狼编码空间与离散问题解空间的转换; 算法提出移动平均自适应灰狼更新策略和灰狼基因遗传策略来提高全局收敛能力; 算法提出带 3-opt 的劣势点启发邻域搜索策略来增强算法的全局和局部搜索能力, 实现 CVRP 的求解.

## 2 问题描述及数学模型

CVRP 用无向图  $G = (V, E)$  来表示,  $V = \{1, \dots, n\}$  为顶点集合, 包括车场和客户, 其中 1 为车场.  $E = \{(i, j) | i, j \in V\}$  为边集, 开销矩阵  $C$  的元素  $c_{ij}$  是边  $(i, j)$  的开销. 客户  $i$  的货物运输需求以  $d_i$  表示. 可用车辆数量为  $K$ , 每台车辆容量限制为  $Q$ .

CVRP 寻找不多于  $K$  条简单回路, 使得总开销最低, 并且满足每个客户只能由 1 辆车提供运输服务, 每台车辆以车场为出发点和返回点, 车辆累计运输量不超过车辆容量限制.

定义决策变量为:

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 经过边 } (i, j) \\ 0, & \text{其他} \end{cases} \quad (1)$$

$$y_{ik} = \begin{cases} 1, & \text{车辆 } k \text{ 访问顶点 } i \\ 0, & \text{其他} \end{cases} \quad (2)$$

建立 CVRP 的数学模型如下所示:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \quad (3)$$

$$\text{s. t. } \sum_{k=1}^K y_{ik} = 1, \quad \forall i \in V \setminus \{1\} \quad (4)$$

$$\sum_{k=1}^K y_{1k} \leq K \quad (5)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \quad \forall i \in V, k = 1, \dots, K \quad (6)$$

$$\sum_{i \in V} d_i y_{ik} \leq Q, \quad \forall k = 1, \dots, K \quad (7)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad \forall S \subseteq V \setminus \{1\}, |S| \geq 2, \quad k = 1, \dots, K \quad (8)$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in V; k = 1, \dots, K \quad (9)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in V; k = 1, \dots, K \quad (10)$$

其中, 式(3)为目标函数, 表示总开销最小. 式(4)、式(5)表示客户仅由 1 辆车提供服务, 车场出发车辆不多于  $K$ . 式(6)保证每个顶点车辆到达和离开的数量平衡. 式(7)为车辆容量约束. 式(8)消除子回路. 式(9)和式(10)表示决策变量为 0-1 变量.

## 3 自适应遗传灰狼优化算法

### 3.1 离散灰狼优化算法

Mirjalili 等<sup>[8]</sup>提出的 GWO 是在连续域内进行寻优, 本文提出的 AGGWOA 中将灰狼个体位置和客户集的序列进行映射, 对客户序列进行全局和局部寻优, 获取最佳配送路径, 实现 CVRP 的求解.

(1) 灰狼优化算法基本思想

灰狼分为  $\alpha, \beta, \delta, \omega$  四个等级, 适应度最好的三匹狼为  $\alpha, \beta, \delta$ , 其余为  $\omega$ ,  $\omega$  依据  $\omega$  自身当前位置和  $\alpha, \beta, \delta$  的位置来更新位置. 位置更新如式(13)、式(14)、式(15).

$$A = 2a \cdot r_1 - a \quad (11)$$

$$C = 2 \cdot r_2 \quad (12)$$

$$\begin{aligned} D_\alpha &= |C \cdot X_\alpha(t) - X(t)| \\ D_\beta &= |C \cdot X_\beta(t) - X(t)| \\ D_\delta &= |C \cdot X_\delta(t) - X(t)| \end{aligned} \quad (13)$$

$$\begin{aligned} X_1 &= X_\alpha - A_1 \cdot (D_\alpha) \\ X_2 &= X_\beta - A_2 \cdot (D_\beta) \\ X_3 &= X_\delta - A_3 \cdot (D_\delta) \end{aligned} \quad (14)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (15)$$

其中,  $t$  为当前迭代,  $r_1$  和  $r_2$  为  $[0, 1]$  范围的随机向量,  $a$  中分量在迭代中从 2 线性减少到 0.  $X_\alpha(t)$ 、 $X_\beta(t)$ 、 $X_\delta(t)$  为  $\alpha, \beta, \delta$  当前位置,  $X(t)$  为  $\omega$  自身当前位置,  $X(t+1)$  为  $\omega$  下一步的新位置,  $A_1, A_2, A_3$  计算如式(11). 当  $a$  的分量小

于 1 时,灰狼攻击猎物,当  $a$  的分量大于 1 时,灰狼进行狩猎。 $C$  控制灰狼在以头狼为中心的一个随机大小的范围内搜索猎物。GWO 详细描述见文献[8]。

### (2) 离散灰狼优化算法改进

本文 AGGWOA 与 GWO 具有相同的领导层级和灰狼行为,并进行了灰狼空间整数编码改进,编解码如第 3.2 节所述。

#### (a) 跟随操作定义

为实现离散域灰狼跟随头狼狩猎、攻击猎物和搜寻猎物等行为,需要重新定义灰狼跟随操作。设头狼为  $\alpha$ ,群狼为  $\omega$ ,位置向量为  $m$  维整数向量。如果使  $\omega$  的  $l$  个连续位置分量序列与  $\alpha$  的  $l$  个连续位置分量序列相同,则认为  $\omega$  跟随  $\alpha$  移动  $l$ 。可知  $\omega$  与  $\alpha$  的最大可移动距离为  $m$ ,这时  $\omega$  与  $\alpha$  分量序列完全相同。设相同位置分量序列在  $\alpha$  中的起始序号为  $o_s$ ,在  $\omega$  中为  $o_d$ ,若  $o_s$  不等于  $o_d$ ,则认为  $\omega$  趋近目标是  $\alpha$  为中心的给定范围内的一个点;若  $o_s$  等于  $o_d$ ,灰狼沿直线趋近头狼;若  $|o_s - o_d| = m - l$ ,则  $\omega$  趋近目标是  $\alpha$  为中心的给定范围的边缘。各变量实现灰狼个体各动作如图 1 所示。

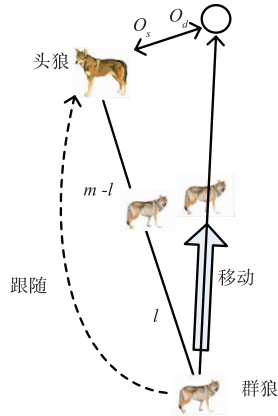


图1 各变量实现灰狼个体各动作示意图

#### (b) 位置更新操作

群狼  $\omega$  位置更新如式(18)到式(22)所示。

$$A = 2a \cdot r_1 - a \quad (16)$$

$$C = 2 \cdot r_2 \quad (17)$$

$$l = \lceil m \times (1 - \text{abs}(A/2)) \rceil \quad (18)$$

$$o_s = \lceil r_3 \times (m - l) \rceil \quad (19)$$

$$o_d = \begin{cases} \lceil C \times o_s \rceil, & \lceil C \times o_s \rceil \leq (m - l) \\ \lceil C \times o_s \rceil - (m - l), & \text{其他} \end{cases} \quad (20)$$

$$X_i^{\text{new}} = \begin{cases} X_\alpha(o_s : o_s + l), & r_4 \geq \psi_1 \\ X_\beta(o_s : o_s + l), & \psi_1 > r_4 \geq \psi_2 \\ X_\delta(o_s : o_s + l), & r_4 < \psi_2 \end{cases} \quad (21)$$

$$X_i^{t+1} = \lambda(X_i^{\text{new}}, l, X_i^t, o_d) \quad (22)$$

其中,  $r_1, r_2, r_3$  和  $r_4$  为  $[0, 1]$  范围的随机数,  $X_\alpha^t, X_\beta^t, X_\delta^t$  分

别为  $\alpha, \beta, \delta$  当前位置,  $X_i^t$  为灰狼  $i$  当前位置,  $X_i^{t+1}$  为灰狼  $i$  下一步的更新位置。[ ] 表示四舍五入取整,  $X_\alpha^t(o_s : o_s + l), X_\beta^t(o_s : o_s + l), X_\delta^t(o_s : o_s + l)$  分别表示  $X_\alpha^t, X_\beta^t, X_\delta^t$  第  $o_s$  到  $o_s + l$  的分量组成的向量。  $\lambda(X_i^{\text{new}}, l, X_i^t, o_d)$  表示维数为  $l$  的  $X_i^{\text{new}}$  并入  $X_i^t$  的位置  $o_d$  中,并消除  $X_i^t$  中原有重复元素。  $\psi_1, \psi_2$  为跟随概率常数,  $\psi_1, \psi_2$  取 0.6, 0.25, 灰狼  $i$  按 4:3:5:2.5 的概率随机取头狼  $\alpha, \beta, \delta$  中的其中一个进行跟随操作。

#### (c) 灰狼行为控制

攻击猎物:  $A$  为  $[-a, a]$  的随机数,  $a = 2$  时,  $l$  很小,在算法内遗传算子的作用下,  $\omega$  远离  $\alpha, \beta, \delta$ ;  $a$  逐步减少,  $l$  逐步接近  $m$ ,  $\omega$  逐渐趋近头狼  $\alpha, \beta, \delta$ , 体现灰狼攻击猎物行为。

搜寻猎物: 变量  $C$  是 0 到 2 的随机数, 通过  $o_s, o_d$  使  $\omega$  向  $\alpha, \beta$  或  $\delta$  为中心的一定范围内随机位置移动, 从而控制灰狼搜索猎物。

### 3.2 编解码策略

本文 AGGWOA 采用先路由后分组<sup>[10]</sup>的方法生成 CVRP 解决方案; 为了与问题域的邻域搜索和局部搜索结合, 本文算法采用灰狼空间整数编码策略, CVRP 解空间的顶点序号与灰狼空间的位置分量一一对应。具体编解码操作如下:

(1) 编码: 在编码的设计上, 采用只保留客户的连接操作进行编码。具体如下: 设有  $n$  个顶点, 客户数量  $m = n - 1$ 。将所有车辆路径去除车场后首尾相连组成  $m$  维的基本通路向量, 记为  $T$ , 由于 1 为车场, 对  $T$  分量  $t_i$  有  $2 \leq t_i \leq m + 1, \forall i \in [1, m]$ 。定义灰狼的  $m$  维整数位置向量为  $X$ , 对  $X$  分量  $x_i$  有  $1 \leq x_i \leq m, \forall i \in [1, m]$ 。将  $T$  分量  $t_i$  值减 1, 一一映射为灰狼位置分量  $x_i$ , 将  $T$  映射为灰狼位置  $X$ 。

(2) 解码: 灰狼  $m$  维位置向量  $X$  的每个分量加 1, 一一映射为  $T$  中分量, 将灰狼位置映射为包含全部客户的基本通路向量。采用 Prins<sup>[11]</sup> 提出的最优分割过程分组的方法, 通过  $[T, Q, d, C]$  四元组对基本通路向量进行分组, 其中  $T$  为灰狼位置得到的基本通路向量,  $Q$  为车辆最大载重容量,  $d = (d_1, d_2, \dots, d_n)$  是客户需求向量,  $C$  是顶点间开销矩阵, 顶点包括车场和所有客户。解码的步骤主要由计算最佳分割点和生成最优分割的路径分组组成, 最优分割过程描述见文献[11]。解码得到的所有路径分组首尾均为车场。

### 3.3 适应度函数

适应度函数是用来对群体进行评价, 作为选举头狼和遗传选择的依据。本文令适应度函数为:

$$\text{fit} = \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \quad (23)$$

### 3.4 移动平均自适应灰狼更新策略

如第 3.1 节所描述, GWO 通过控制变量  $a$  控制灰

狼搜寻猎物和攻击猎物行为. 本文算法提出了一种移动平均自适应灰狼更新策略. 当算法运行超过 20 次迭代以后, 每轮迭代均按式 (24) 计算除当前迭代的最近 20 次迭代的最优解适应度平均值作为移动平均值  $\sigma$ .

$$\sigma = \frac{\left( \sum_{i=NT-20}^{NT-1} f_i \right)}{20} \quad (24)$$

其中 NT 是当前迭代数,  $f_i$  第  $i$  次迭代的最优解适应度. 式 (25) 在控制参数  $a$  更新计算时附加移动平均值  $\sigma$  和当前最佳适应度的比值关联的增量.

$$a = \min \left\{ 2 - NT \frac{2}{NT_{\max}} + r \left( \frac{f_{NT-1}}{\sigma} \right)^s, 2 \right\}, \quad 0 \leq r < 2, s \in N^+ \quad (25)$$

其中  $NT_{\max}$  是最大迭代数,  $f_{NT-1}$  是上一轮迭代的最优解适应度,  $r, s$  分别为自适应比重系数和自适应调节因子. 算法可感知当前迭代得到的全局最优值比最近若干次迭代得到的全局最优值的变化, 控制灰狼的行为, 从而达到探索和围猎的自适应交替进行, 避免 CVRP 解空间搜索陷入局部最优.

### 3.5 灰狼基因遗传策略

为了增强 GWO 求解 CVRP 的全局搜索能力, 本文提出了一种灰狼基因遗传策略. 该策略以灰狼位置向量为遗传算法个体基因, 引入遗传算法的选择操作和变异操作来实现灰狼个体或群体的位置变化来增强算法搜索能力, 同时使用停滞重置策略来避免算法的“早熟”问题.

(1) 选择操作: 采用最佳保留复制法, 将灰狼个体按适应度由最优到最差排序, 通过群体总数与选择比例因子  $P_s$  计算复制个体数量, 将该个体数量的最优灰狼基因逐个复制到相同数量的最差灰狼个体基因中, 以加快算法的收敛.

(2) 变异操作: 采用交换变异方法, 当产生随机数小于变异概率因子  $P_m$  时进行变异操作, 以增加多样性, 避免算法陷入局部最优.

(3) 停滞重置策略: 令 TL 为最优解停滞迭代次数, 初始值为 0.1 次迭代结束时未得到新最优解时,  $TL = TL + 1$ , 否则,  $TL = 0$ ; 如果 TL 大于 50 且为 10 的整数倍, 则将适应度后 50% 的灰狼个体的位置向量随机初始化.

### 3.6 劣势点启发邻域搜索策略

本文提出劣势点启发邻域搜索策略, 在构建的超路径上先采用评判值轮盘赌方法来选择弱势点, 再选择适应度变化最佳的顶点与弱势点进行客户抽出-插入操作或客户交换操作, 生成新的路径, 再对新路径进行 3-opt 局部搜索操作.

#### 3.6.1 构建超路径

为了实现本文劣势点启发邻域搜索策略, 构建车辆超路径: 将所有车辆路径分组  $L^k, k \in [1, K]$  首尾相连, 相连处只保留一个车场, 形成  $m + K + 1$  维向量  $L$ , 包含  $m$  个客户和  $K + 1$  个车场.  $L$  总开销等于所有  $L^k$  开销之和.

#### 3.6.2 局部搜索操作

基于我们之前的研究<sup>[12-14]</sup>证明: 局部搜索策略能提高算法的计算精度和收敛速度. 因此, 劣势点启发邻域搜索策略引入 3-opt 局部搜索操作, 局部搜索方式是对超路径中的单车路径逐个进行. 3-opt 具体步骤见文献<sup>[15]</sup>.

#### 3.6.3 劣势点启发邻域搜索策略执行流程

下文中用变量  $L(x)$  和  $L'(x)$  来分别表示超路径向量  $L$  和  $L'$  中的第  $x$  个分量, 其值为  $V$  的元素. 劣势点启发邻域搜索策略执行流程如下:

##### 算法 1 劣势点启发邻域搜索策略

输入: 车辆路径分组  $L^k, k \in [1, K]$ , 车辆最大载重容量  $Q$ , 客户需求向量  $d$ , 开销矩阵  $C$ ;  
输出: 车辆路径分组  $L^k, k \in [1, K]$

1. 根据 3.6.1 节构造超路径  $L$ ;
2. foreach  $i$  in  $[1, m + K + 1]$
3. 根据式 (27) 计算  $L(i)$  顶点客户抽出-插入操作评判值  $v_i$ ;
4. end foreach
5. 根据式 (29)、式 (30) 执行轮盘赌选择, 确定劣势点序号  $j$ ;
6. 构造将劣势点  $L(j)$  移除后的新超路径向量  $L' = L \setminus \{L(j)\}$ ;
7. foreach  $p$  in  $[1, m + K]$
8. 根据式 (31) 计算在  $L'$  向量的第  $p$  顶点后插入  $L(j)$  情况下路程开销增加值  $a_p$ ; 如果插入后路径违反容量约束, 则  $a_p = M$ ,  $M$  为大整数;
9. end foreach
10. 根据式 (32) 选取最优操作位置  $w$ ; 对劣势点  $L(j)$  在  $L'$  向量最优操作位置  $w$  进行插入操作, 得到新超路径  $L''$ ; 对  $L''$  根据 3.6.2 节进行局部搜索;
11.  $L = L''$
12. foreach  $i$  in  $[1, m + K + 1]$
13. 根据式 (28) 计算  $L(i)$  顶点客户交换评判值  $v_i$ ;
14. end foreach
15. 根据式 (29)、式 (30) 执行轮盘赌选择, 确定劣势点序号  $j$ ;
16. foreach  $p$  in  $[1, m + K + 1]$
17. 根据式 (26) 计算  $L$  向量的第  $j$  顶点与  $L$  向量的第  $p$  顶点交换情况下路程开销变化值  $a_p$ ; 如果交换后路径违反容量约束, 则  $a_p = M$ ,  $M$  为大整数;
18. end foreach
19. 根据式 (32) 选取最优操作位置  $w$ ; 对  $L(j)$  和  $L(w)$  进行客户交换操作, 得到新超路径  $L'$ ; 对  $L'$  根据 3.6.2 节进行局部搜索;
20. 将  $L'$  转换成  $K$  个单车路径分组.

算法 1 中应用的公式如下:

$$a_p = \begin{cases} c_{L(j-1),L(p)} + c_{L(p),L(j+1)} \\ \quad + c_{L(p-1),L(j)} + c_{L(j),L(p+1)} \\ \quad - (c_{L(j-1),L(j)} + c_{L(j),L(j+1)} \\ \quad + c_{L(p-1),L(p)} + c_{L(p),L(p+1)}), \\ \quad |j-p| > 2, L(p) \neq 1 \\ c_{L(j-1),L(p)} + c_{L(j),L(p+1)} \\ \quad - (c_{L(j-1),L(j)} + c_{L(p),L(p+1)}), \\ \quad p = j+1, L(p) \neq 1 \\ 0, \\ \quad p = j, L(p) \neq 1 \\ c_{L(p-1),L(j)} + c_{L(p),L(j+1)} \\ \quad - (c_{L(p-1),L(p)} + c_{L(j),L(j+1)}), \\ \quad p = j-1, L(p) \neq 1 \\ M, \\ \quad L(p) = 1 \end{cases} \quad (26)$$

$$v_i = \begin{cases} (c_{L(i-1),L(i)} + c_{L(i),L(i+1)} \\ \quad - c_{L(i-1),L(i+1)})^2, L(i) \neq 1 \\ 0, \\ \quad L(i) = 1 \end{cases} \quad (27)$$

$$v_i = \begin{cases} (c_{L(i-1),L(i)} + c_{L(i),L(i+1)})^2, & L(i) \neq 1 \\ 0, & L(i) = 1 \end{cases} \quad (28)$$

$$R = U[0, \sum_{i=1}^{m+K+1} v_i] \quad (29)$$

$$j = x, s. t. \sum_{i=1}^x v_i \leq R < \sum_{i=1}^{x+1} v_i \text{ 且 } v_x \neq 0 \quad (30)$$

$$a_p = c_{L(p),L(j)} + c_{L(j),L(p+1)} - c_{L(p),L(p+1)} \quad (31)$$

$$w = \arg \min_p (a_p) \quad (32)$$

其中  $U[0, \sum_{i=1}^{m+K+1} v_i]$  表示在区间  $[0, \sum_{i=1}^{m+K+1} v_i]$  服从均匀分布的随机数;

### 3.7 算法步骤

AGGWOA 的算法步骤如算法 2:

#### 算法 2 AGGWOA

输入: 最大迭代次数  $NT_{\max}$ , 灰狼数量  $NG$ , 自适应比重系数  $r$ , 自适应调节因子  $s$ , 选择比例因子  $P_s$ , 变异概率因子  $P_m$ , 车辆最大载重容量  $Q$ , 客户需求向量  $d$ , 开销矩阵  $C$ , 顶点数  $n$ ;

输出: 当前最优路径

1. 随机生成灰狼种群  $X_i (i = 1, 2, \dots, NG)$ , 头狼  $X_\alpha, X_\beta, X_\delta$  初始化为空, 初始化当前迭代变量  $NT = 0$ , 当前最优路径为空;
2. while  $NT < NT_{\max}$
3. if  $NT \neq 0$  then
4. 对当前最优路径执行  $\sqrt{n}$  次第 3.6 节劣势点启发邻域搜索策略操作; 更新当前最优路径和头狼个体  $X_\alpha$  位置向量;
5. end if
6. foreach WF in  $[1, NG]$
7. 对  $X_{WF}$  位置向量按照第 3.2 节进行解码操作得到当前车辆路径; 对当前车辆路径执行第 3.6 节劣势点启发邻域搜索策略操作; 根据新路径更新  $X_{WF}$  位置向量; 按第 3.3 节计算  $X_{WF}$  路径适应度;

8. 依次比较  $X_{WF}$  和头狼个体  $X_\alpha, X_\beta, X_\delta$  的适应度, 如较优则  $X_{WF}$  选举为对应头狼; 如果  $X_{WF}$  选举为  $X_\alpha$ , 对当前车辆路径执行  $\sqrt{n}$  次第 3.6 节劣势点启发邻域搜索策略操作; 更新当前最优路径和头狼个体  $X_\alpha$  位置向量;
9. end foreach
10. 对灰狼种群的位置编码按第 3.5 节进行灰狼基因选择操作和变异操作, 执行停滞重置策略;
11. 按第 3.1、3.4 节更新灰狼种群当前位置;  $NT = NT + 1$ ;
12. end while

## 4 实验与分析

### 4.1 实验环境

本文的所有实验硬件环境为处理器 Intel Core i5-6600 3.30GHz(4 核) 1 个, 内存 8GB RAM, 编程运行环境为 MATLAB 2017a. 算例取自经典 CVRP 算例的 Set A 和 Set P. 路网的边的开销采用边的两个顶点的欧氏距离四舍五入取整得到.

### 4.2 参数实验与分析

本算法参数有第 3.4 节的  $r, s$  和第 3.5 节的  $P_s, P_m$ , 其中  $r, s$  用于计算自适应灰狼更新策略中移动平均值和控制变量  $a$ , 影响的是灰狼群体整体行为;  $P_s, P_m$  是遗传策略中的比例和概率, 该策略在单次迭代中更新局部个体. 根据参数间关联紧密程度分为  $r, s$  和  $P_s, P_m$  两组; 根据先整体后局部的原则, 参数实验先对  $r, s$  组合进行分析, 再对  $P_s, P_m$  组合进行测试分析. 取 A-n33-k5、A-n69-k9、B-n41-k6、B-n68-k9、P-n50-k10、P-n70-k10 等 6 个算例每组参数运算 10 次, 统计最小值和平均值, 计算平均值偏差 = (平均值 - 最优解) / 最优解, 最小值偏差 = (最小值 - 最优解) / 最优解.

每一个参数组合 6 个算例的最小值偏差和平均值偏差的累加计为该参数组合的最小值偏差和平均值偏差. 计算每一个组合最小值偏差占比 = 每一个组合最小值偏差 / 所有组合最小值偏差合计; 同理, 计算每一个组合的平均值偏差占比. 计算加权和 = 最小值偏差占比  $\times 40\%$  + 平均值偏差占比  $\times 60\%$ . 加权和较低表明该参数的效果较好.

#### (1) 自适应更新操作参数分析

设置  $P_s = 20\%$ 、 $P_m = 0.2$ ,  $NG = 40$ ,  $NT_{\max} = 1000$ ,  $r$  取值  $(0, 0.5, 0.9, 1.3, 1.7)$ ,  $s$  取值  $(3, 4, 5, 6)$ .  $r$  取值 0 时算法采用传统步进式更新, 以对比本文自适应灰狼更新策略, 这时  $s$  取值对结果无影响.  $r = 0$  的参数组合仅设 1 组,  $r$  为其它值有 16 组, 实验组成 17 组  $r, s$  参数组合进行测试. 结果如表 1 所示.  $r = 1.3, s = 5$  时得到的加权和为 4.74 最低, 即该组合较优.  $r = 0$  时, 加权和为 7.44, 在所有组合中最高, 表明了自适应灰狼更新策略的有效性. 以下实验取  $r = 1.3, s = 5$ .

表 1 6 个算例汇总  $r$ 、 $s$  取值组合测试结果表

$r$	$s$	最小值 偏差/%	最小值偏 差占比/%	平均值 偏差/%	平均值偏 差占比/%	加权和
0	-	14.02	7.76	27.24	7.24	7.44
0.5	3	13.16	7.28	23.03	6.12	6.58
0.5	4	8.36	4.62	21.97	5.84	5.35
0.5	5	10.02	5.55	23.49	6.24	5.96
0.5	6	14.10	7.80	24.50	6.51	7.03
0.9	3	10.48	5.80	21.93	5.83	5.82
0.9	4	7.67	<b>4.24</b>	22.40	5.95	5.27
0.9	5	10.35	5.73	21.91	5.82	5.78
0.9	6	10.34	5.72	19.74	5.25	5.44
1.3	3	10.90	6.03	23.98	6.37	6.23
1.3	4	7.91	4.38	21.14	5.62	5.12
1.3	5	8.39	4.64	18.12	<b>4.81</b>	<b>4.74</b>
1.3	6	13.29	7.35	22.68	6.03	6.56
1.7	3	7.92	4.38	20.08	5.34	4.95
1.7	4	12.32	6.82	22.27	5.92	6.28
1.7	5	11.64	6.44	22.04	5.86	6.09
1.7	6	9.88	5.47	19.85	5.27	5.35
合计		181.75		379.37		

## (2) 遗传算子参数分析

取  $r = 1.3$ ,  $s = 5$ ,  $NG = 40$ ,  $NT_{\max} = 1000$ ,  $P_s$  取值 (0%, 20%, 40%, 60%),  $P_m$  取值 (0, 0.2, 0.4, 0.6), 组成 16 组参数进行测试. 结果如表 2 所示.  $P_s = 40\%$ ,  $P_m = 0$  的加权和为 5.24 最低, 即该组合最优;  $P_s = 20\%$ ,  $P_m = 0.2$  的加权和为 5.37 次低, 最小值偏差占比最低, 表示该参数在部分算例的寻优方面有一定优势. 实验表明遗传算子提高算法计算结果的质量, 和灰狼优化算法结合效果良好.

## 4.3 对比实验与分析

为了验证 AGGWOA 的 CVRP 求解能力, 与近年的群智能算法和仿生优化算法进行经典 CVRP 算例求解的横向比较. 按照本文第 4.2 节中分析, 取  $r = 1.3$ ,  $s = 5$ ,  $NG = 320$ ,  $NT_{\max} = 1000$ ,  $P_s = 40\%$ ,  $P_m = 0$ , 每个算例运

行 20 次. 表格中运算结果的最小值与其他算法结果最小值进行逐个算例横向比较, 最优的以加粗表示, 计算结果最小值达到最优解的加星号表示.

表 2 6 个算例汇总  $P_s$ 、 $P_m$  取值组合测试结果表

$P_s$	$P_m$	最小值 偏差/%	最小值偏 差占比/%	平均值 偏差/%	平均值偏 差占比/%	加权和
0%	0	14.92	8.16	26.71	7.30	7.65
0%	0.2	13.72	7.50	25.67	7.02	7.21
0%	0.4	15.34	8.39	27.19	7.44	7.82
0%	0.6	13.70	7.49	29.30	8.01	7.80
20%	0	11.79	6.45	20.60	5.64	5.96
20%	0.2	8.64	<b>4.72</b>	21.22	5.80	5.37
20%	0.4	10.79	5.90	23.61	6.46	6.23
20%	0.6	10.37	5.67	21.48	5.88	5.79
40%	0	10.17	5.56	18.39	<b>5.03</b>	<b>5.24</b>
40%	0.2	10.63	5.81	19.52	5.34	5.53
40%	0.4	9.38	5.13	22.26	6.09	5.70
40%	0.6	11.42	6.25	21.91	5.99	6.09
60%	0	9.58	5.24	20.38	5.57	5.44
60%	0.2	10.96	6.00	21.04	5.75	5.85
60%	0.4	11.24	6.15	23.26	6.36	6.28
60%	0.6	10.20	5.58	23.11	6.32	6.02
合计		182.85		365.65		

## (1) Set A 算例下求解能力分析与比较

取 Set A 中的全部 27 个算例, 对比 Adaptive Sweep + VTPSO<sup>[5]</sup>、LNS-ACO<sup>[3]</sup>、EBMC-ABC<sup>[4]</sup>、SC-ESA<sup>[16]</sup>、HVNSOS<sup>[7]</sup>, 对本文算法进行算法效果分析, 测试结果如表 3. 从算法的最小值偏差统计来看, AGGWOA (0.30%) 优于 Adaptive Sweep + VTPSO<sup>[5]</sup> (10.46%) 和 LNS-ACO<sup>[3]</sup> (0.60%), 比 EBMC-ABC<sup>[4]</sup> (0.19%)、SC-ESA<sup>[16]</sup> (0.16%) 和 HVNSOS<sup>[7]</sup> (0.13%) 略差, 差距很小, 最小值偏差相差分别仅为 0.11%, 0.14%, 0.17%.

表 3 Set A 算例测试结果

算例	已知 最优解	AGGWOA			Adaptive Sweep + VTPSO <sup>[5]</sup>		LNS-ACO <sup>[3]</sup>		EBMC-ABC <sup>[4]</sup>		SC-ESA <sup>[16]</sup>		HVNSOS <sup>[7]</sup>	
		平均值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%
A-n32-k5	784	0.43	<b>784 *</b>	<b>0</b>	882	12.50	<b>784 *</b>	<b>0</b>	<b>784 *</b>	<b>0</b>	<b>784 *</b>	<b>0</b>	<b>784 *</b>	<b>0</b>
A-n33-k5	661	0.08	<b>661 *</b>	<b>0</b>	698	5.60	<b>661 *</b>	<b>0</b>	<b>661 *</b>	<b>0</b>	<b>661 *</b>	<b>0</b>	<b>661 *</b>	<b>0</b>
A-n33-k6	742	0.09	<b>742 *</b>	<b>0</b>	751	1.21	<b>742 *</b>	<b>0</b>	<b>742 *</b>	<b>0</b>	<b>742 *</b>	<b>0</b>	<b>742 *</b>	<b>0</b>
A-n34-k5	778	0.62	<b>778 *</b>	<b>0</b>	785	0.90	<b>778 *</b>	<b>0</b>	<b>778 *</b>	<b>0</b>	<b>778 *</b>	<b>0</b>	<b>778 *</b>	<b>0</b>

续表

算例	已知最优解	AGGWOA			Adaptive Sweep + VTPSO <sup>[5]</sup>		LNS-ACO <sup>[3]</sup>		EBMC-ABC <sup>[4]</sup>		SC-ESA <sup>[16]</sup>		HVNSOS <sup>[7]</sup>	
		平均值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%
A-n36-k5	799	0.94	<b>799</b> *	<b>0</b>	881	10.26	<b>799</b> *	<b>0</b>	<b>799</b> *	<b>0</b>	<b>799</b> *	<b>0</b>	<b>799</b> *	<b>0</b>
A-n37-k5	669	0.54	<b>669</b> *	<b>0</b>	754	12.71	<b>669</b> *	<b>0</b>	<b>669</b> *	<b>0</b>	<b>669</b> *	<b>0</b>	<b>669</b> *	<b>0</b>
A-n37-k6	949	1.20	<b>949</b> *	<b>0</b>	1112	17.18	<b>949</b> *	<b>0</b>	<b>949</b> *	<b>0</b>	<b>949</b> *	<b>0</b>	<b>949</b> *	<b>0</b>
A-n38-k5	730	0.50	<b>730</b> *	<b>0</b>	813	11.37	<b>730</b> *	<b>0</b>	<b>730</b> *	<b>0</b>	<b>730</b> *	<b>0</b>	<b>730</b> *	<b>0</b>
A-n39-k5	822	0.49	<b>822</b> *	<b>0</b>	877	6.69	<b>822</b> *	<b>0</b>	<b>822</b> *	<b>0</b>	<b>822</b> *	<b>0</b>	<b>822</b> *	<b>0</b>
A-n39-k6	831	0.69	833	0.24	972	16.69	<b>831</b> *	<b>0</b>	<b>831</b> *	<b>0</b>	<b>831</b> *	<b>0</b>	<b>831</b> *	<b>0</b>
A-n44-k6	937	1.26	<b>937</b> *	<b>0</b>	1056	12.70	<b>937</b> *	<b>0</b>	<b>937</b> *	<b>0</b>	<b>937</b> *	<b>0</b>	<b>937</b> *	<b>0</b>
A-n45-k6	944	1.88	953	0.95	1073	13.19	958	1.48	949	0.53	<b>944</b> *	<b>0</b>	<b>944</b> *	<b>0</b>
A-n45-k7	1146	1.35	<b>1146</b> *	<b>0</b>	1305	13.87	<b>1146</b> *	<b>0</b>	<b>1146</b> *	<b>0</b>	<b>1146</b> *	<b>0</b>	<b>1146</b> *	<b>0</b>
A-n46-k7	914	1.12	<b>914</b> *	<b>0</b>	975	6.67	<b>914</b> *	<b>0</b>	<b>914</b> *	<b>0</b>	<b>914</b> *	<b>0</b>	<b>914</b> *	<b>0</b>
A-n48-k7	1073	2.40	<b>1073</b> *	<b>0</b>	1152	7.36	1084.1	1.03	<b>1073</b> *	<b>0</b>	1084	1.03	<b>1073</b> *	<b>0</b>
A-n53-k7	1010	2.08	1017	0.69	1090	7.18	<b>1010</b> *	<b>0</b>	<b>1010</b> *	<b>0</b>	1011	0.1	<b>1010</b> *	<b>0</b>
A-n54-k7	1167	2.03	1176	0.77	1361	16.62	<b>1167</b> *	<b>0</b>	<b>1167</b> *	<b>0</b>	1168	0.09	<b>1167</b> *	<b>0</b>
A-n55-k9	1073	1.77	1074	0.09	1190	10.90	<b>1073</b> *	<b>0</b>	<b>1073</b> *	<b>0</b>	<b>1073</b> *	<b>0</b>	<b>1073</b> *	<b>0</b>
A-n60-k9	1354	2.29	1359	0.37	1503	8.99	<b>1354</b> *	<b>0</b>	1355	0.07	1355	0.07	<b>1354</b> *	<b>0</b>
A-n61-k9	1034	1.56	1035	0.09	1164	12.36	1067	3.19	1035	0.09	<b>1034</b> *	<b>0</b>	1035	0.09
A-n62-k8	1288	2.87	1298	0.78	1408	7.73	1308	1.55	1300	0.93	1298	0.78	<b>1291</b>	<b>0.23</b>
A-n63-kl0	1314	2.36	1323	0.68	1823	11.50	1329	1.14	1319	0.38	<b>1315</b>	<b>0.08</b>	1319	0.38
A-n63-k9	1616	2.76	1629	0.80	1477	10.31	1649	2.04	1627	0.68	<b>1624</b>	<b>0.50</b>	1628	0.74
A-n64-k9	1401	2.97	1414	0.93	1598	12.61	1415	1	1412	0.79	<b>1409</b>	<b>0.57</b>	1414	0.93
A-n65-k9	1174	2.05	1178	0.34	1317	11.14	1185	0.94	1178	0.34	1178	0.34	<b>1177</b>	<b>0.26</b>
A-n69-k9	1159	2.20	1163	0.35	1259	7.70	1170	0.95	1166	0.60	<b>1159</b> *	<b>0</b>	<b>1159</b> *	<b>0</b>
A-n80-kl0	1763	3.77	1783	1.13	2136	16.59	1815	2.95	1774	0.62	<b>1776</b>	<b>0.74</b>	1779	0.91
平均		1.57		0.30		10.46		0.60		0.19		0.16		<b>0.13</b>

表 4 Set P 算例测试结果

算例	已知最优解	AGGWOA			Adaptive Sweep + VTPSO <sup>[5]</sup>		K-GWO <sup>[2]</sup>		LNS-ACO <sup>[3]</sup>		SC-ESA <sup>[16]</sup>		HVNSOS <sup>[7]</sup>	
		平均值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%
P-n16-k8	450	<b>0</b>	<b>450</b> *	<b>0</b>	549	22.00	—	—	<b>450</b> *	<b>0</b>	<b>450</b> *	<b>0</b>	<b>450</b> *	<b>0</b>
P-n19-k2	212	<b>0</b>	<b>212</b> *	<b>0</b>	246	16.04	—	—	<b>212</b> *	<b>0</b>	219	3.30	<b>212</b> *	<b>0</b>
P-n20-k2	216	<b>0</b>	<b>216</b> *	<b>0</b>	249	15.28	217	0.46	<b>216</b> *	<b>0</b>	218	0.93	<b>216</b> *	<b>0</b>
P-n21-k2	211	0	<b>211</b> *	<b>0</b>	<b>211</b> *	<b>0</b>	<b>211</b> *	<b>0</b>	<b>211</b> *	0	212	0.47	<b>211</b> *	<b>0</b>
P-n22-k2	216	0	<b>216</b> *	<b>0</b>	<b>216</b> *	<b>0</b>	<b>216</b> *	<b>0</b>	<b>216</b> *	<b>0</b>	<b>216</b> *	<b>0</b>	<b>216</b> *	<b>0</b>
P-n22-k8	603	-2.16	<b>590</b> *	<b>-2.16</b>	633	4.98	—	—	—	—	—	—	<b>603</b> *	<b>0</b>
P-n23-k8	529	0	<b>529</b> *	<b>0</b>	634	19.85	—	—	<b>529</b> *	<b>0</b>	<b>529</b> *	<b>0</b>	<b>529</b> *	<b>0</b>

续表

算例	已知最优解	AGGWOA			Adaptive Sweep + VTPSO <sup>[5]</sup>		K-GWO <sup>[2]</sup>		LNS-ACO <sup>[3]</sup>		SC-ESA <sup>[16]</sup>		HVNSOS <sup>[7]</sup>	
		平均值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%	结果 最小值	最小值 偏差/%
P-n40-k5	458	0.79	<b>458 *</b>	<b>0</b>	483	5.46	—	—	<b>458 *</b>	<b>0</b>	459	0.22	<b>458 *</b>	<b>0</b>
P-n45-k5	510	0.74	<b>510 *</b>	<b>0</b>	524	2.75	<b>510 *</b>	<b>0</b>	<b>510 *</b>	<b>0</b>	511	0.20	<b>510 *</b>	<b>0</b>
P-n50-k7	554	1.76	<b>554 *</b>	<b>0</b>	583	5.23	—	—	<b>554 *</b>	<b>0</b>	<b>554 *</b>	<b>0</b>	<b>554 *</b>	<b>0</b>
P-n50-k8	631	1.35	<b>631 *</b>	<b>0</b>	677	7.29	—	—	643	1.9	637	0.95	632	0.16
P-n50-k10	696	3.25	704	1.15	783	12.50	—	—	<b>696 *</b>	<b>0</b>	697	0.14	<b>696 *</b>	<b>0</b>
P-n51-k10	741	2.33	743	0.27	802	8.23	—	—	747	0.81	<b>741 *</b>	<b>0</b>	744	0.40
P-n55-k7	568	2.09	570	0.35	595	4.75	—	—	<b>568 *</b>	<b>0</b>	574	1.06	<b>568 *</b>	<b>0</b>
P-n55-k8	588	-0.01	<b>576 *</b>	<b>-2.04</b>	586	-0.34	606	3.06	—	—	—	—	—	—
P-n55-k10	694	1.74	696	0.29	745	7.35	—	—	<b>694 *</b>	<b>0</b>	695	0.14	698	0.58
P-n60-k10	744	3.16	753	1.21	830	11.56	—	—	755	1.48	<b>745</b>	<b>0.13</b>	748	0.54
P-n60-k15	968	3.53	976	0.83	1119	15.60	—	—	977	0.93	<b>968 *</b>	<b>0</b>	<b>968 *</b>	<b>0</b>
P-n76-k4	593	1.80	<b>593 *</b>	<b>0</b>	612	3.20	614	3.54	—	—	—	—	—	—
P-n76-k5	627	2.89	<b>628</b>	<b>0.16</b>	647	3.19	635	1.28	—	—	—	—	—	—

## (2) Set P 算例下求解能力分析与比较

取 Set P 中 20 个算例,对比 Adaptive Sweep + VTPSO 算法<sup>[5]</sup>、K-GWO 算法<sup>[2]</sup>、LNS-ACO 算法<sup>[3]</sup>、SC-ESA 算法<sup>[16]</sup>、HVNSOS 算法<sup>[7]</sup>,对本文算法进行算法效果分析,测试结果如表 4. AGGWOA 优于上述 5 个算法的算例数量分别为 18、4、4、7、4,而上述 5 个算法优于 AGGWOA 算例数量分别为 0、0、3、5、4. AGGWOA 输出结果优化程度优于 Adaptive Sweep + VTPSO<sup>[5]</sup>、K-GWO<sup>[2]</sup>、LNS-ACO<sup>[3]</sup>和 SC-ESA<sup>[16]</sup>,与 HVNSOS<sup>[7]</sup>持平。

在对比实验中,本文算法在求解 CVRP 问题方面具有较好的性能,整体上优于 Adaptive Sweep + VTPSO<sup>[5]</sup>、K-GWO<sup>[2]</sup>、LNS-ACO<sup>[3]</sup>、EBMC-ABC<sup>[4]</sup>和 SC-ESA<sup>[16]</sup>. 算法在客户均匀分布、客户聚集、单车路径较多客户、单车路径少量客户等特征的算例的求解上均表现优异,算法输出结果稳定。

## 5 结论

本文提出了一种自适应遗传灰狼优化算法(AGGWOA),实现 CVRP 的求解,灰狼优化算法经离散化改进后,编码空间和解空间转换较为简单直接,容易与问题域的邻域搜索和局部搜索结合,算法扩展性较好.通过本文算法和其他算法对 Set A 和 Set P 算例集的求解对比实验,证明了本文算法的有效性和实用性. 本文研究为 CVRP 问题的求解提供新的求解思路,丰富了其求解方法. 灰狼优化算法在多车场、多车型、带时间窗的 VRP 问题求解将是下一步的研究方向。

## 参考文献

- [1] Dantzig G B, Ramser J H. The truck dispatching problem [J]. Management Science, 1959, 6(1): 80-91.
- [2] Korayem L, Khorsid M, Kassem S S. Using grey wolf algorithm to solve the capacitated vehicle routing problem [J]. IOP Conference Series: Materials Science and Engineering, 2015, 83(1): 012014.
- [3] Akpinar S. Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem [J]. Expert Systems with Applications, 2016, 61: 28-38.
- [4] Ng K K H, Lee C K M, Zhang S Z, et al. A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion [J]. Computers & Industrial Engineering, 2017, 109: 151-168.
- [5] Akhand M A H, Peya Z J, Murase K. Capacitated vehicle routing problem solving using adaptive sweep and velocity tentative PSO [J]. International Journal of Advanced Computer Science and Applications, 2017, 8(12): 288-295.
- [6] Hannan M A, Akhtar M, Begum R A, et al. Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm [J]. Waste Management, 2018, 71: 31-41.
- [7] 李阳, 范厚明. 求解带容量约束车辆路径问题的混合变邻域生物共栖搜索算法 [J]. 控制与决策, 2018, 33(07): 1190-1198.

- LI Yang, FAN Hou-ming. Hybrid variable neighborhood symbiotic organisms search for capacitated vehicle routing problem [J]. Control and Decision, 2018, 33 (07): 1190 – 1198. (in Chinese)
- [8] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46 – 61.
- [9] Faris H, Aljarah I, Al-Betar M A, et al. Grey wolf optimizer: a review of recent variants and applications [J]. Neural Computing and Applications, 2017, 30 (2): 413 – 435.
- [10] Beasley J E. Route first—Cluster second methods for vehicle routing [J]. Omega, 1983, 11 (4): 403 – 408.
- [11] Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem [J]. Computers & Operations Research, 2004, 31 (12): 1985 – 2002.
- [12] 戚远航, 蔡延光, 蔡颢, 等. 旅行商问题的混沌混合离散蝙蝠算法 [J]. 电子学报, 2016, 44 (10): 2543 – 2547.  
QI Yuan-hang, CAI Yan-guang, CAI Hao, et al. Chaotic hybrid discrete bat algorithm for traveling salesman problem [J]. Acta Electronica Sinica, 2016, 44 (10): 2543 – 2547. (in Chinese)
- [13] 戚远航, 蔡延光, 蔡颢, 等. 带时间窗的车辆路径问题的离散蝙蝠算法 [J]. 电子学报, 2018, 46 (3): 672 – 679.  
QI Yuan-hang, CAI Yan-guang, CAI Hao, et al. Discrete bat algorithm for vehicle routing problem with time window [J]. Acta Electronica Sinica, 2018, 46 (3): 672 – 679. (in Chinese)
- [14] 戚远航, 蔡延光, 蔡颢, 等. 泰森多边形的离散蝙蝠算法求解多车场车辆路径问题 [J]. 控制理论与应用, 2018, 35 (8): 1142 – 1150.  
QI Yuan-hang, CAI Yan-guang, CAI Hao, et al. Voronoi diagram-based discrete bat algorithm for multi-depot vehicle routing problem [J]. Control Theory & Applications, 2018, 35 (8): 1142 – 1150. (in Chinese)
- [15] Helsgaun K. General k-opt submoves for the Lin-Kernighan TSP heuristic [J]. Mathematical Programming Computation, 2009, 1 (2): 119 – 163.
- [16] Stanojevic M, Stanojevic B, Vujosevic M. Enhanced savings calculation and its applications for solving capacitated vehicle routing problem [J]. Applied Mathematics & Computation, 2013, 219 (20): 10302 – 10312.

#### 作者简介



**黄戈文** 男, 1972 年 7 月出生于广东梅县. 现为广东工业大学自动化学院博士生, 嘉应学院信息网络中心系统分析师. 研究方向为组合优化、运输调度、计算智能等.

E-mail: huang\_gewen@163.com



**蔡延光 (通信作者)** 男, 1963 年 2 月出生于湖北咸宁. 1988 年和 1996 年分别在重庆大学和浙江大学获理学硕士和工学博士学位. 现为广东工业大学教授, 博士生导师, 从事复杂网络系统建模、控制与优化、物流控制与优化、智能交通系统、组合优化、智能优化、物联网信息处理与优化控制等方面的研究.

E-mail: caiyg99@163.com



**戚远航** 男, 1993 年 6 月出生于广东湛江. 2018 年在广东工业大学获得工学博士学位, 现为电子科技大学中山学院讲师, 从事复杂系统建模与优化、智能规划、运输调度的研究.

E-mail: qiyuanhang77@163.com