

基于改进鬣狗优化算法的多阈值彩色图像分割

贾鹤鸣 姜子超 彭晓旭 康立飞 李瑶 孙康健

(东北林业大学机电工程学院 黑龙江 哈尔滨 150040)

摘要 针对传统鬣狗优化算法在处理图像分割问题时容易陷入局部最优、收敛效率低等问题,提出基于混沌初始化策略、非线性收敛因子调整策略、莱维飞行策略以及精英反向学习策略的改进鬣狗优化算法(Improved Spotted Hyena Optimizer, ISHO),并应用于多阈值彩色图像分割问题中。该算法不仅提升了在解决复杂性问题时的寻优效率,而且提高了求解精度与质量,避免了受随机因子影响而陷入局部最优的缺陷。为了验证该算法的有效性,利用伯克利图像分割验证,结果表明:该算法在分割速度及效率上具有明显优势,求解精度高,具有较好的工程实用性。

关键词 鬣狗算法 彩色图像分割 多阈值 混沌初始化 精英反向学习 非线性收敛 莱维飞行

中图分类号 TP391 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2020.05.045

MULTI-THRESHOLD COLOR IMAGE SEGMENTATION BASED ON IMPROVED SPOTTED HYENA OPTIMIZER

Jia Heming Jiang Zichao Peng Xiaoxu Kang Lifei Li Yao Sun Kangjian

(College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, Heilongjiang, China)

Abstract In order to solve the problems of local optimization and low convergence efficiency in traditional spotted hyena optimizer, we propose an improved spotted hyena optimizer based on chaos initialization strategy, non-linear convergence factor adjustment strategy, Levy flight strategy and elite reverse learning strategy. And it is applied to multi-threshold color image segmentation. It not only improves the efficiency of the algorithm in solving complex problems, but also improves the accuracy and quality of the solution. It avoids the defect of falling into local optimum under the influence of random factors. In order to verify the effectiveness of our algorithm, Berkeley image segmentation is used to verify the results. The results show that our algorithm has obvious advantages in segmentation speed and efficiency, high solution accuracy and good engineering practicability.

Keywords Spotted hyena optimizer Color image segmentation Multi-threshold Chaotic initialization Elite opposition-based learning Non-linear convergence Levy flight

0 引言

图像分割是图像分析的关键步骤,是图像标识和描述的必要前提和基础,因此分割方法的选择和分割精度是极为重要的^[1]。现有的分割算法多基于边缘、区域、阈值及特定理论等。阈值图像分割法运算效率高、性能较稳定,是目前最广泛应用的一种分割方法。阈值分割法的基本理论是依据某一准则求出最佳阈

值,并进行像素点比较,将目标和背景区域区分出来。单阈值分割基本思想是按阈值把直方图分割成目标和背景两类,而多阈值分割则是将图像分成多个类,使得多个类的类间方差最大^[2]。因为单阈值分割图像效果不够理想,所以近些年研究学者多采用迭代的方式将其研究领域延伸至多阈值分割。

经典 Otsu 算法本质上是进行穷举搜索,如果采用穷举法进行多阈值的搜索,总计算量较大。利用类间方差最大化思想,可以将图像的多阈值分割问题转化

为求最优阈值的函数优化问题,因此,为缩短运算时间,提升效率,可选用算法优化思想进行多阈值选取过程。但由于优化算法本身的局限性,在分割图像时仍存在求解质量低、容易陷入局部最优等问题。

鬣狗算法 (Spotted Hyena Optimizer, SHO) 是印度塔帕尔大学 Gaurav Dhiman 教授于 2018 年基于非洲斑点鬣狗种群觅食机制提出的一种新型群智能优化算法,主要通过模拟斑点鬣狗种群的捕食行为特性来达到优化搜索的目的。SHO 算法操作简单、设置参数少、稳定性强、求解精度更高,因其性能良好,已经开始被应用于解决非线性连续优化问题等多个工程领域。但由于其全局搜索能力和局部搜索能力仅通过 B 和 E 两个随机参数进行动态调节,且这两个阶段种群位置更新公式未进行明显区分,SHO 算法不可避免地存在易陷入局部最优解、求解质量低等类似缺陷,而在复杂优化问题中,大多要求优化算法尽可能遍历到全局最优解可能的分布范围^[3]。所以 SHO 仍存在局部最优、复杂优化效果不良等缺陷。

针对标准 SHO 算法受随机因子的影响而导致的全局优化效果差、收敛效率不良、求解速率与质量低等问题,本文提出一种改进鬣狗算法 ISHO,并将其应用到多阈值图像分割领域。通过引入混沌初始化策略、非线性收敛因子调整策略、莱维飞行策略以及精英反向学习策略,来提高算法在解决复杂性问题时的寻优效率及求解精度。为检验鬣狗优化算法及改进的鬣狗优化算法在图像多阈值分割方面的可行性及分割精度的优越性,选取 6 幅经典伯克利图像进行多阈值分割实验,利用分割时间、适应度值、PSNR 值、FISM 值等指标评价分割效果,对比粒子群优化算法 (particle swarm optimization, PSO)、正余弦优化算法 (sine-cosine optimization algorithm, SCA)、灰狼优化算法 (grey wolf optimization algorithm, GWO)、鲸鱼优化算法 (whale optimization algorithm, WOA)、标准 SHO 等五种优化算法,实验结果表明:ISHO 算法在分割速度及效率上具有明显优势,且图像失真度较小,分割效果较好,有利于图像提取的后续处理。

1 多阈值 Otsu 图像分割

在过去的几十年中,Otsu 的类间方法以其无监督的自动阈值选择技术而闻名^[4]。类概率 ω_k 、类平均水平 μ_k 和总平均水平 μ_T 分别表示为:

$$\omega_k = \sum_{j=t_k}^{t_{k+1}-1} P_j \quad (1)$$

$$\mu_k = \sum_{j=t_k}^{t_{k+1}-1} \frac{jP_j}{\omega_k} \quad (2)$$

$$\mu_T = \sum_{j=0}^{L-1} jP_j \quad (3)$$

则目标函数可以描述为:

$$Fit = \sum_{k=0}^K \omega_k (\mu_k - \mu_T)^2 \quad (4)$$

Otsu 最佳阈值的定义为:

$$(t_1^*, t_2^*, \dots, t_k^*) = \underset{0 < t_1 < t_2 < \dots < t_k < L-1}{\operatorname{argmax}} (Fit(t_1, t_2, \dots, t_k)) \quad (5)$$

2 鬣狗优化算法 (SHO)

鬣狗种群的捕食机制包括搜索、包围、狩猎和攻击猎物四个过程^[5]。鬣狗算法的基本原理如下:

(1) 包围机制:鬣狗具有熟悉并判断猎物的位置,从而包围它们的能力。该行为的数学模型为:

$$D_h = |B \cdot P_p(t) - P(t)| \quad (6)$$

$$B = 2 \cdot r_1 \quad (7)$$

式中: D_h 表示鬣狗个体与猎物之间的距离; t 表示当前迭代次数; P_p 表示猎物的位置; $P(t)$ 表示鬣狗种群中的个体位置; B 表示摇摆因子; r_1 表示 $[0, 1]$ 之间均匀分布的随机数。

鬣狗的个体位置更新如下所示:

$$P(t+1) = P_p(t) - E \cdot D_h \quad (8)$$

$$E = 2 \cdot h \cdot r_2 - h \quad (9)$$

$$h = 5 - 5 \cdot \frac{\text{Iteration}}{NI} \quad (10)$$

式中: E 表示收敛因子; r_2 表示 $[0, 1]$ 之间均匀分布的随机数; h 表示控制因子,随迭代次数的增加线性递减,变化范围为 $[0, 5]$; Iteration 表示算法运行当前迭代数目,其中 $\text{Iteration} = 1, 2, \dots, NI$; NI 表示最大迭代次数。

(2) 狩猎机制:斑点鬣狗通常依靠可信赖的种群网络及识别猎物位置的能力来生活和分组捕杀。该机制的具体描述如下:

$$D_h = |B \cdot P_p(t) - P_k| \quad (11)$$

$$P_k = P_h - E \cdot D_h \quad (12)$$

$$C_h = P_k + P_{k+1} + \dots + P_{k+N} \quad (13)$$

式中: P_h 定义了第一个最佳斑点鬣狗的位置; P_k 表示其他斑点鬣狗的位置; N 表示斑点鬣狗的数量; C_h 是 N 个最优解的集群。其中 N 计算如下:

$$N = \text{Count}_{\text{nos}}(P_h, P_{h+1}, P_{h+2}, \dots, (P_h + M)) \quad (14)$$

式中: M 是 $[0.5, 1]$ 中的随机向量,在添加 M 之后, nos 定义可行解的数量并计算所有候选解,其与给定搜索空间中的最优解相似。

(3) 攻击猎物(局部搜索):鬣狗在猎食的最后阶

段开始攻击猎物,当收敛因子 $|E| < 1$ 时,鬣狗个体便会向猎物发动攻击。全局最优解通过求取当前最优解集的平均值来确定鬣狗搜索个体的更新趋势。攻击猎物的数学公式具体描述如下:

$$P_h(t+1) = \frac{C_h}{N} \quad (15)$$

式中: $P_h(t+1)$ 为保存最优解; C_h 表示最优解群集。

(4) 搜索机制(全局探索):鬣狗大多根据位于最优解群集 C_h 中的鬣狗群或群集的位置来搜寻猎物,当收敛因子 $|E| > 1$ 时,鬣狗会分散并远离当前猎物,搜索更适合的猎物的位置,这种机制使得SHO算法能够在全局进行搜索。

3 改进鬣狗优化算法(ISHO)

3.1 混沌初始化策略

混沌理论因为具有强大的遍历性、敏感性和非重复性而被广泛引入群智能算法中用以增强种群多样性以改善其优化性能。与随机搜索相比,它可以以更高的概率和速度对搜索空间进行彻底搜索。目前群智能算法文献多采用Logistic映射产生混沌序列,但均匀性较差。相比Logistic映射,Tent映射结构简单,迭代速度快,具有更好的遍历均匀性^[6]。

因此,本文在鬣狗算法迭代初期进行混沌初始化,选用Tent映射产生的混沌序列来改进SHO算法,基于混沌Tent映射理论产生鬣狗种群的初始位置,使其均匀地分布在搜索空间内,以增加个体的多样性,增大跳出局部最优的概率和对解空间进行深入搜索的能力,有助于提高求解效率和质量^[7]。Tent映射数学表达式为:

$$y_{i+1} = \begin{cases} 2y_i & 0 \leq y_i \leq 0.5 \\ 2(1-y_i) & 0.5 \leq y_i \leq 1 \end{cases} \quad (16)$$

经伯努利位移变换后可得:

$$y_{i+1} = (2y_i) \bmod 1 \quad (17)$$

3.2 非线性收敛因子调整策略

标准鬣狗优化算法的控制因子 h 采用的是线性递减策略,但是算法迭代过程实质上是非线性变化的,控制因子 h 线性递减的方式不能准确地体现出实际的优化搜索过程。因此,本文引入一种新的递减方式,具体描述如下所示:

$$h(t) = 5 - 5 \left[\frac{1}{e - 1} (e^{\frac{t}{NI}} - 1) \right]^Q \quad (18)$$

式中: t 为当前迭代次数; NI 表示最大迭代次数; Q 表示衰减系数,取值范围为 $[0, 10]$ 并取整, Q 越大,对应的控制因子 h 衰减越剧烈。改进后的控制因子 h 随着

迭代次数从5到0呈非线性递减。在迭代初期,改进后的控制因子 h 递减缓慢,呈渐进搜索状态,衰减性优于原始的控制因子,便于更好地寻找全局最优解。在后期,改进的控制因子 h 的衰减程度开始提高,搜索速度加快,并向最优解靠近,有助于提高搜索到的局部最优解的精确性。因此,这种非线性收敛因子更符合算法的实际收敛过程,进一步增强了算法全局寻优能力^[8]。

3.3 莱维飞行策略

莱维飞行策略属于一类非高斯随机过程,其特点是随机性比较强,飞行的步长满足一个重尾的稳定分布。在飞行过程中,前期大步长后期小步长,不仅扩大了种群的多样性,也使种群在小范围内可以更好地收敛于全局最优解。莱维飞行位置更新公式为:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(s) \quad i = 1, 2, \dots, n \quad (19)$$

式中: $x_i^{(t)}$ 表示 x_i 第 t 代位置; \oplus 表示点乘; α 表示步长控制量; $Levy(s)$ 为随机搜索路径,表示服从参数为 s 的莱维分布。即:

$$Levy(s) \sim u = t^{-1-\beta} \quad (20)$$

式中: s 表示步长,指数变量取值范围一般为 $0 < \beta \leq 2$,本文取值 $\beta = 1.5$ 。

$$s \sim \frac{u}{|v|^{\frac{1}{\beta}}} \quad (21)$$

式中: u, v 均服从标准正态分布:

$$\begin{cases} u \sim N(0, \delta_u^2) \\ v \sim N(0, \delta_v^2) \end{cases} \quad (22)$$

$$\begin{cases} \delta_u = \left[\frac{\Gamma(1+\beta) \times \sin \frac{\pi\beta}{2}}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}} \\ \delta_v = 1 \end{cases} \quad (23)$$

式中: Γ 为积分运算,是标准的Gamma函数。

研究表明,莱维飞行可以避免出现局部徘徊现象,引入莱维飞行,可以扩大种群的多样性以及种群搜索范围,有助于算法跳出局部最优,提高算法的寻优能力,从而达到提高全局最优的目的^[9]。因此,为了增强鬣狗个体之间的信息交流,将莱维飞行策略引入鬣狗优化算法中,并作用于随机因子 r_1 和 r_2 中。鬣狗优化算法引入莱维飞行策略后的包围机制具体描述如下:

$$D_h = |B \cdot P_p(x) - P(x)| \quad (24)$$

$$P(x+1) = P_p(x) - E \cdot D_h \quad (25)$$

摇摆因子:

$$B = 2 \cdot Levy \cdot r_1 \quad (26)$$

收敛因子:

$$E = 2 \cdot h \cdot Levy \cdot r_2 - h \quad (27)$$

$$h = 5 - 5 \cdot \frac{\text{Iteration}}{NI} \quad (28)$$

3.4 精英反向学习策略

Tizhoosh 在 2005 年提出反向学习策略 (Opposition-Based Learning, OBL), 具有良好的性能, 至今已应用于多个领域。由于在生成反向解过程中缺乏方向引导机制, 受粒子群优化 (PSO) 思想的启发, 周新宇等在反向学习策略的基础上提出了精英反向学习策略 (Elite Opposition-Based Learning, EOBL), 精英反向学习策略可以通过利用当前群体中的最优解来引导反向解生成过程, 利用精英个体丰富的信息和反向种群来增加种群多样性^[10-11]。EOBL 算法具体描述如下: 假设当前鬣狗种群中精英向量 (当前最优解) 为 $x_i = (x_1, x_2, \dots, x_N)$, 其中 $x_j \in (LB_j, UB_j)$, 反向解表示为 $x'_i = (x'_1, x'_2, \dots, x'_N)$, 此时原始解和反向解各维度对应关系为:

$$x'_j = \alpha \cdot (LB_j + UB_j) - x_j \quad (29)$$

式中: 常数因子 $\alpha \in U(0, 1)$; (LB_j, UB_j) 表示第 j 个决策变量的动态边界。

$$LB_j = \min(x_{i,j}) \quad (30)$$

$$UB_j = \max(x_{i,j}) \quad (31)$$

式中: $\min(x_{i,j})$ 、 $\max(x_{i,j})$ 分别表示鬣狗种群中第 j 维个体的最小值和最大值。

在精英反向学习策略中, 可通过对常数因子 α 随机取值生成多个不同的反向精英鬣狗个体来构造反向精英鬣狗种群, 这样有利于搜索反向精英鬣狗个体所处的领域空间, 提高算法的局部搜索能力。当反向解超出范围 (LB_j, UB_j) 失效时, 需要在边界内生成一个随机位置来进行越界复位操作, 具体描述为:

$$x_j = \text{rand}(0, 1) \cdot (LB_j, UB_j) \quad (32)$$

算法执行过程中, EOBL 策略择优保留最优鬣狗个体, 不代表每一次迭代都会进行更新。重复无筛选地进行反向学习, 会增加计算量, 影响算法收敛速度, 此时最优鬣狗个体更新效果会下降。因此, 本文引入执行精英反向学习的概率 p_1 对反向学习的执行前提加以有效控制, 取值为 0.6。改进后的鬣狗优化算法基本思想是: 当随机数小于 p_1 设置值时, 则执行精英反向学习策略, 反之则不执行。此时的新解更新机制具体描述如下:

$$x_{\text{new},j} = \begin{cases} x_{\text{new},j} = LB_j + UB_j - x_j & \text{rand} < p_1 \\ x_{\text{new},j} = \frac{Ch}{N} & p_1 < \text{rand} < 1 \end{cases} \quad (33)$$

式中: rand 表示 $[0, 1]$ 上均匀分布的随机数。

3.5 算法描述

改进的鬣狗算法流程图如图 1 所示。

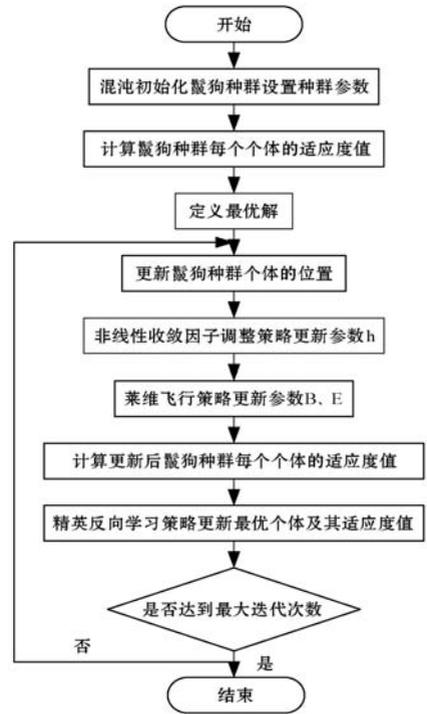


图 1 改进鬣狗优化算法

4 实验

4.1 实验背景与设计

采用改进鬣狗优化算法结合 Otsu 多阈值分割思想对伯克利彩色图像进行有效处理, 并将该方法与 PSO、SCA、GWO、SHO 及 WOA 等算法的分割效果进行对比, 验证所选改进算法的可行性。

(1) 峰值信噪比 (PSNR)。计算比较分割后图像与原图的 PSNR 值, PSNR 值越大, 则失真越少, 说明该算法在多阈值分割应用上的性能越好^[12-13]。PSNR 计算过程如下:

$$MSE = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N (I(i,j) - \hat{I}(i,j))^2}{M \times N}} \quad (34)$$

$$PSNR = 20 \lg \left(\frac{255}{MSE} \right) \quad (35)$$

式中: MSE 表示原图像与分割后图像之间的均方误差; $I(i,j)$ 、 $\hat{I}(i,j)$ 分别表示尺寸为 $M \times N$ 的原图像和分割后的图像。

(2) 特征相似性 (FSIM)。特征相似性为一种基于图像的相位一致性与图像的空域梯度特征来评价图像质量的指标^[14]。

$$FSIM = \frac{\sum_{x \in X, y \in Y} S_L(x,y) PC_m(x,y)}{\sum_{x \in X, y \in Y} PC_m(x,y)} \quad (36)$$

式中: $S_L(x,y)$ 用来评价图像的相似性; $PC_m(x,y)$ 表示

相位相似性; $x \in X, y \in Y$ 表示图像是整个空域。

为减小算法随机性的影响,提高实验的公平性,实验仿真环境均设置为 Windows7 系统,Intel(R) pentium(R) CPU G3260,内存均为 3.30 GHz,仿真软件为 MATLAB R2014b。实验中测试函数决策变量的维数均设置为 30,最大迭代次数设置为 10 000,种群规模设置为 30。所有算法的参数设置如表 1 所示。

表 1 算法参数设置

算法	参数	取值
PSO ^[15]	c_1, c_2	1.5
	ω	0.686
	ν	[0,1]
	a	2
SCA ^[16]	r_1	[0,2n]
	r_2	[-2,2]
	r_3	[0,1]
WOA ^[17]	b	1
	a	[0,2]
	l	[-1,1]
SHO	h	[0,5]
ISHO	M	[0.5,1]
	β	1.5
GWO ^[18]	p_1	0.6
	a	[0,2]

4.2 伯克利经典图像分割实验

为检验鬣狗优化算法及改进的鬣狗优化算法在图像多阈值分割方面的可行性及分割精度的优越性,选取 6 幅伯克利经典图像进行多阈值分割实验,采用优化算法对最佳阈值分割点进行寻优,获得每幅图像的最佳阈值及其他评价指标值,并与 PSO、SCA、GWO、WOA、SHO 等算法进行比较。通过算法的分割时间、最佳适应度值、PSNR 值和 FSIM 值四种性能指标来评估 ISHO 算法的性能与其他五种的差异。伯克利测试图像如图 2 所示。

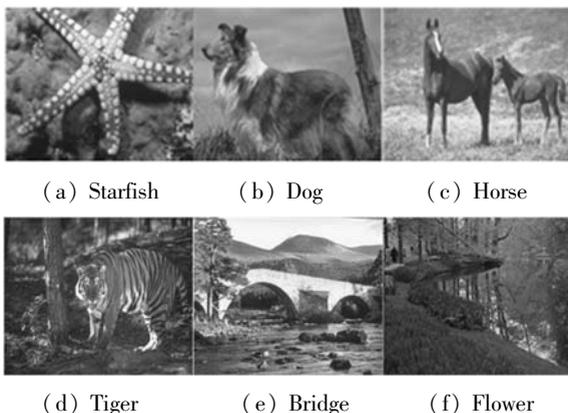


图 2 伯克利经典图像

所有图像分割后的实验结果依次为阈值 $n = \{2, 3, 4, 5\}$ 时采用 PSO、SCA、GWO、WOA、SHO 及 ISHO 算法分割后的结果,表 2、表 3、表 4 及表 5 分别为基于 Otsu 的 PSO、SCA、GWO、WOA、SHO 及 ISHO 算法的分割时间、最佳适应度值、PSNR 值和 FSIM 值。

表 2 各算法分割时间

图像	n	Time					
		PSO	SCA	GWO	WOA	SHO	ISHO
Starfish	2	1.218 042	1.107 050	0.415 062	0.791 492	0.707 112	0.779 409
	3	1.676 360	1.288 001	0.453 537	0.837 965	0.757 271	0.841 641
	4	1.962 434	1.522 744	0.513 141	0.887 222	0.800 674	0.893 762
Dog	2	2.044 308	2.062 692	0.562 800	0.936 777	0.852 270	0.940 428
	3	1.192 799	1.224 310	0.404 080	0.780 891	0.711 160	0.785 216
	4	1.472 483	1.315 142	0.459 648	0.834 346	0.784 329	0.840 264
Horse	2	1.748 292	1.351 972	0.519 046	0.883 329	0.808 508	0.896 620
	3	2.082 583	1.556 137	0.571 361	0.927 360	0.862 336	0.937 576
	4	1.203 334	1.291 532	0.400 701	0.791 699	0.712 165	0.781 520
Tiger	2	1.532 573	1.320 629	0.454 371	0.838 462	0.756 069	0.835 192
	3	1.886 435	1.552 707	0.506 613	0.905 342	0.811 095	0.889 563
	4	1.988 662	2.059 304	0.559 261	0.937 917	0.852 630	0.934 688
Bridge	2	1.249 190	1.236 399	0.408 994	0.781 606	0.712 100	0.797 520
	3	1.514 292	1.431 156	0.464 851	0.835 770	0.765 752	0.826 993
	4	1.741 856	1.567 165	0.511 740	0.893 277	0.807 040	0.878 508
Flower	2	2.018 289	1.738 859	0.566 486	0.938 272	0.856 514	0.943 333
	3	1.208 802	1.163 603	0.398 873	0.783 279	0.720 992	0.794 226
	4	1.473 643	1.310 642	0.474 297	0.843 345	0.756 896	0.833 750
Dog	2	1.742 179	1.471 794	0.511 654	0.894 576	0.806 382	0.886 588
	3	2.085 756	2.011 740	0.561 756	0.940 757	0.863 757	0.938 344
	4	1.196 751	1.160 561	0.415 074	0.775 285	0.702 499	0.783 172
Horse	2	1.475 980	1.330 393	0.462 514	0.836 248	0.758 395	0.840 466
	3	1.741 473	1.559 663	0.532 536	0.888 774	0.805 846	0.888 301
	4	2.003 636	1.689 115	0.574 614	0.954 742	0.856 010	0.965 635

表 3 各算法最佳适应度值

图像	n	适应度值					
		PSO	SCA	GWO	WOA	SHO	ISHO
Starfish	2	2.810 5e+03	2.810 5e+03	2.810 5e+03	2.805 1e+03	2.796 4e+03	2.800 8e+03
	3	2.985 7e+03	2.985 7e+03	2.985 7e+03	2.967 1e+03	2.972 7e+03	2.931 3e+03
	4	3.071 0e+03	3.071 0e+03	3.071 0e+03	3.036 5e+03	3.047 9e+03	3.011 7e+03
Dog	2	3.114 6e+03	3.114 6e+03	3.114 6e+03	3.087 8e+03	3.087 8e+03	3.066 4e+03
	3	2.106 8e+03	2.106 8e+03	2.106 8e+03	2.098 9e+03	2.100 3e+03	2.099 1e+03
	4	2.199 8e+03	2.199 8e+03	2.199 8e+03	2.180 6e+03	2.182 2e+03	2.175 6e+03
Horse	2	2.250 0e+03	2.250 0e+03	2.250 0e+03	2.214 2e+03	2.225 1e+03	2.217 9e+03
	3	2.281 5e+03	2.281 5e+03	2.281 4e+03	2.248 8e+03	2.238 4e+03	2.260 6e+03
	4	1.765 3e+03	1.765 3e+03	1.765 3e+03	1.752 4e+03	1.760 5e+03	1.753 6e+03
Flower	2	1.895 4e+03	1.895 4e+03	1.895 4e+03	1.855 5e+03	1.885 7e+03	1.886 0e+03
	3	1.960 2e+03	1.960 2e+03	1.960 2e+03	1.920 0e+03	1.931 4e+03	1.908 3e+03
	4	1.996 7e+03	1.996 7e+03	1.996 6e+03	1.964 6e+03	1.967 5e+03	1.946 0e+03

续表 3

图像	n	适应度值					
		PSO	SCA	GWO	WOA	SHO	ISHO
Tiger	2	1.353 5e+03	1.353 5e+03	1.353 5e+03	1.352 5e+03	1.342 8e+03	1.347 8e+03
	3	1.468 5e+03	1.468 5e+03	1.468 5e+03	1.443 2e+03	1.441 5e+03	1.451 7e+03
	4	1.526 6e+03	1.526 6e+03	1.526 6e+03	1.511 3e+03	1.494 5e+03	1.500 6e+03
	5	1.560 0e+03	1.560 0e+03	1.560 0e+03	1.520 0e+03	1.531 4e+03	1.538 7e+03
Bridge	2	4.089 6e+03	4.089 6e+03	4.089 6e+03	4.082 0e+03	4.071 7e+03	4.085 8e+03
	3	4.301 7e+03	4.301 7e+03	4.301 7e+03	4.281 2e+03	4.255 2e+03	4.280 3e+03
	4	4.386 0e+03	4.386 0e+03	4.386 0e+03	4.325 5e+03	4.311 3e+03	4.326 3e+03
	5	4.431 5e+03	4.431 5e+03	4.431 4e+03	4.393 2e+03	4.370 5e+03	4.393 3e+03
Flower	2	1.047 9e+03	1.047 9e+03	1.047 9e+03	1.045 1e+03	1.045 2e+03	1.044 7e+03
	3	1.140 8e+03	1.140 8e+03	1.140 8e+03	1.119 7e+03	1.108 9e+03	1.129 7e+03
	4	1.191 7e+03	1.191 7e+03	1.191 7e+03	1.170 2e+03	1.161 8e+03	1.167 2e+03
	5	1.214 5e+03	1.214 5e+03	1.214 5e+03	1.198 3e+03	1.196 3e+03	1.202 0e+03

表 4 各算法 PSNR 值

图像	n	PSNR					
		PSO	SCA	GWO	WOA	SHO	ISHO
Starfish	2	15.412 3	15.421 3	15.427 5	15.362 6	15.294 8	15.505 6
	3	17.922 7	17.922 7	17.922 7	17.123 1	17.574 9	17.039 2
	4	19.917 6	19.917 6	19.920 6	19.479 0	19.269 0	18.896 8
	5	21.292 1	21.292 1	21.291 7	20.511 5	21.057 6	20.009 4
Dog	2	16.348 1	16.348 1	16.348 1	15.253 1	16.450 1	16.120 4
	3	17.743 6	17.743 6	17.743 6	17.396 0	17.747 1	17.998 4
	4	19.934 6	19.934 6	19.984 5	20.058 9	18.362 7	20.214 6
	5	21.210 7	21.210 7	21.280 1	20.091 9	20.017 8	21.069 4
Horse	2	15.094 7	15.094 7	15.094 7	14.938 1	15.007 9	15.081 6
	3	16.831 7	16.831 7	16.831 7	16.155 4	16.126 8	16.630 1
	4	18.376 0	18.375 2	18.375 2	18.103 4	17.376 0	17.545 5
	5	19.523 8	19.523 8	19.546 9	18.197 3	20.089 6	17.667 6
Tiger	2	14.747 6	14.747 6	14.747 6	14.752 7	14.901 1	15.140 1
	3	17.021 2	17.021 2	17.021 2	16.853 9	16.220 1	17.622 2
	4	18.798 7	18.798 7	18.798 7	18.621 2	20.103 1	19.063 5
	5	20.459 0	20.459 0	20.458 3	20.043 6	20.973 1	19.151 4
Bridge	2	14.149 6	14.149 6	14.149 6	13.965 4	14.107 5	13.966 6
	3	16.624 9	16.624 9	16.624 9	16.744 2	16.133 3	16.481 7
	4	18.768 4	18.768 4	18.768 4	17.090 4	17.373 9	18.580 0
	5	20.252 0	20.252 0	20.267 2	19.705 4	18.356 8	19.091 9
Flower	2	17.736 9	17.736 9	17.736 9	17.629 8	17.696 8	17.666 8
	3	19.414 8	19.414 8	19.414 8	19.938 7	18.653 8	19.716 0
	4	21.653 0	21.653 0	21.670 2	21.101 6	20.944 4	21.996 8
	5	23.139 6	23.631 8	23.636 2	22.639 0	22.729 8	23.155 1

表 5 各算法 FSIM 值

图像	n	FSIM					
		PSO	SCA	GWO	WOA	SHO	ISHO
Starfish	2	0.736 3	0.736 3	0.736 3	0.735 9	0.709 3	0.734 6
	3	0.804 2	0.804 2	0.804 2	0.797 2	0.791 7	0.787 9
	4	0.845 2	0.845 2	0.845 4	0.831 8	0.836 3	0.802 3
	5	0.878 5	0.878 5	0.878 3	0.850 3	0.872 5	0.903 4
Dog	2	0.769 0	0.769 0	0.769 0	0.773 8	0.776 5	0.761 4
	3	0.795 2	0.795 2	0.795 2	0.798 4	0.810 7	0.799 3
	4	0.839 9	0.839 9	0.841 6	0.822 6	0.827 4	0.861 4
	5	0.867 3	0.867 3	0.868 9	0.848 7	0.828 9	0.915 2
Horse	2	0.764 8	0.764 8	0.764 8	0.764 9	0.761 2	0.761 7
	3	0.822 4	0.822 4	0.822 4	0.822 4	0.822 7	0.830 4
	4	0.863 9	0.863 5	0.863 5	0.844 7	0.866 9	0.831 3
	5	0.888 1	0.888 1	0.888 1	0.863 9	0.873 0	0.895 7
Tiger	2	0.727 0	0.727 0	0.727 0	0.727 5	0.732 6	0.732 1
	3	0.789 1	0.789 1	0.789 1	0.782 4	0.800 7	0.810 9
	4	0.832 8	0.832 8	0.832 8	0.850 7	0.835 6	0.808 9
	5	0.860 0	0.863 1	0.863 1	0.876 6	0.862 4	0.895 9
Bridge	2	0.759 7	0.759 7	0.759 7	0.753 1	0.762 2	0.758 6
	3	0.828 0	0.828 0	0.828 0	0.844 4	0.838 7	0.834 8
	4	0.876 9	0.876 9	0.876 9	0.850 5	0.841 4	0.869 5
	5	0.901 1	0.901 1	0.901 5	0.893 2	0.882 8	0.903 8
Flower	2	0.760 8	0.760 8	0.760 8	0.765 6	0.772 9	0.752 8
	3	0.805 0	0.805 0	0.805 0	0.850 7	0.808 2	0.818 2
	4	0.863 2	0.863 2	0.863 4	0.853 8	0.871 9	0.884 7
	5	0.902 7	0.910 9	0.904 2	0.905 0	0.884 0	0.902 1

4.3 实验结果分析

(1) 分割时间:对比分析表 2 的算法分割时间可以发现,6 种算法的分割时间在阈值数目相同时,GWO 算法分割时间最短,WOA、SHO 和 ISHO 算法时间相近,PSO 算法耗费时间远长于其他算法。随着阈值数目的增加,计算量增大,算法复杂度呈指数次幂增长趋势,运行时间也随之增加,当阈值数目 $n = 3、4、5$ 时,PSO 算法所用时间明显增加,依旧大于其他算法,ISHO 算法分割时间和效率仍保持一定优势。实验表明,同 PSO、SCA、GWO、WOA 及 SHO 算法相比,ISHO 算法在分割速度及效率上具有明显优势,能够快速完成彩色图像多阈值分割的任务。

(2) 适应度值:由表 3 可知,分割阈值数目相同时,6 种算法的求解质量(适应度值)相近,其中 PSO 和 ISHO 的适应度值差距极小,求得的最优阈值基本相同。

(3) PSNR 值:结合表 4 可知,当阈值数目 $n = 2$ 时,所有算法的 PSNR 值基本一致,分割后图像效果基本一致,差别不明显,随着阈值数目的增多,当阈值数

目 n 等于 3、4、5 时,6 种算法的 PSNR 值大小顺序为 ISHO > SHO > GW0 > SCA > PSO > WOA, ISHO 算法的分割性能优于其他算法。

(4) FSIM 值:由表 5 可知,随着阈值的增加,6 种算法分割后图像的 FSIM 数值均逐渐增加,ISHO 算法分割后的特征相似性指标 FSIM 数值较大,说明分割后的图像与原图的相似度较高,图像失真度较小,分割效果较好。

实验结果表明,改进鬣狗优化算法能取得与粒子群等群智能算法等同的分割效果,在阈值较多时不易早熟,稳定性更好。与 PSO、SCA、GW0、SHO 及 WOA 算法相比,收敛速度更快,阈值寻优质量较高,且不易陷入局部最优,彩色图像多阈值分割效果较好,分割后的目标图像更加适合后续的图像处理。

5 结 语

鬣狗优化算法具有原理简单、收敛速度快等优点,同时也存在诸多不足,如在高维复杂优化问题时存在全局搜索能力差、易早熟,以及传统多阈值图像分割方法中存在的阈值选择不准确、求解效率低等问题。针对这些不足,本文通过引入混沌初始化策略、非线性收敛因子调整策略、莱维飞行策略及精英反向学习策略,提出了改进的鬣狗优化算法,并通过伯克利经典图像分割测试,验证改进后算法的高效性和可行性。实验结果表明 ISHO 具有较强的工程实用性。

参 考 文 献

[1] 郎春博,贾鹤鸣,邢致恺,等. 基于改进正余弦优化算法的多阈值图像分割[J]. 计算机应用研究, 2019, 37(4): 1-7.

[2] 燕红文,邓雪峰. OTSU 算法在图像分割中的应用研究[J]. 农业开发与装备, 2018(11): 103, 108.

[3] Dhiman G, Kaur A. Spotted hyena optimizer for solving engineering design problems[C]//International Conference on Machine Learning & Data Science. IEEE, 2018.

[4] 邢致恺,贾鹤鸣,宋文龙. 基于莱维飞行樽海鞘群优化算法的多阈值图像分割[J/OL]. 自动化学报: 1-15 [2019-02-23]. <https://doi.org/10.16383/j.aas.c180140>.

[5] Dhiman G, Kumar V. Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems[J]. Knowledge-Based Systems, 2018, 150: 175-197.

[6] Pan Z. An image segmentation approach based on chaotic ant colony algorithms[C]//Proceedings of SPIE—The International Society for Optical Engineering, 2007.

[7] Iacomi M, Cascio D, Fauci F, et al. Mammographic images segmentation based on chaotic map clustering algorithm[J]. BMC Medical Imaging, 2014, 14(1): 12.

[8] 王涛,Chellali R. 非线性权重和收敛因子的鲸鱼算法[J]. 微电子学与计算机, 2019, 36(1): 11-15.

[9] 徐坤,陈志军,闫学勤. 基于莱维飞行的改进蚁群算法求解 TSP 问题[J]. 计算机工程与设计, 2019, 40(1): 245-249.

[10] 汪慎文,丁立新,谢承旺,等. 应用精英反向学习策略的混合差分演化算法[J]. 武汉大学学报, 2013, 59(2): 111-116.

[11] 张超. 一种精英反向学习的花授粉算法[J]. 西安工程大学学报, 2017, 31(6): 847-856.

[12] 刘春燕. 图像分割评价方法研究[D]. 西安: 西安电子科技大学, 2011.

[13] 张晓琳,刘直芳,代金波,等. 基于视觉感知的梯度结构相似度图像质量评价[J]. 计算机应用研究, 2011, 28(6): 2348-2351.

[14] 张媛. 基于结构相似度的图像质量评价技术研究[D]. 西安: 西安电子科技大学, 2014.

[15] 周晓伟,葛永慧. 基于粒子群优化算法的最大类间方差多阈值图像分割[J]. 测绘科学, 2010, 35(2): 88-89, 122.

[16] 鲍小丽,贾鹤鸣,郎春博,等. 正余弦优化算法在多阈值图像分割中的应用[J]. 森林工程, 2019, 35(4): 62-68.

[17] 胡加鑫,贾鹤鸣,邢致恺,等. 基于鲸鱼算法的森林火灾图像多阈值分割[J]. 森林工程, 2018, 34(4): 70-74, 95.

[18] 白媛,陈京荣,展之婵. 改进灰狼优化算法的研究与分析[J]. 计算机科学与应用, 2017, 7(6): 1-5.

(上接第 209 页)

[12] 张哲,孙瑾,杨刘涛. 融合相关滤波与关键点匹配的跟踪算法研究[J]. 光学学报, 2019, 39(2): 259-267.

[13] Wu Y, Lim J, Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.

[14] Bertinetto L, Valmadre J, Golodetz S, et al. Staple: Complementary learners for real-time tracking[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, 2016: 1401-1409.

[15] Danelljan M, Hager G, Khan F S, et al. Learning spatially regularized correlation filters for visual tracking[C]//Proceedings of the 2015 IEEE International Conference on Computer Vision and Pattern Recognition. IEEE Press, 2015: 4310-4318.

[16] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]//Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, 2013: 2411-2418.