

基于改进非洲秃鹫优化算法的图像阈值分割研究

张啸宇, 方忠庆, 杜 义, 孔维宾, 王玉婷, 程子耀

(盐城工学院信息工程学院, 江苏 盐城 224051)

✉ xiaoyuzhang721@163.com; fangzq@ycit.edu.cn; Lemond213@163.com; kongweibin@ycit.cn;
2251193251@qq.com; 3417747952@qq.com



摘 要:针对非洲秃鹫算法(AVOA)全局搜索能力不足与局部搜索策略冗杂的缺点,提出一种改进非洲秃鹫算法(β -PAVOA)。算法采用分段线性混沌映射(PWLCM)初始化种群,增强种群多样性。引入 β 分布与基于饥饿率的搜索策略,增强算法全局搜索能力。改进原算法局部搜索策略,帮助算法及时跳出局部最优。通过8个测试函数验证算法的有效性,并将其应用于二维Otsu图像阈值分割模型。实验结果表明,在测试函数上, β -PAVOA相比较于非洲秃鹫算法(AVOA)、金豺狼优化算法(GJO)、灰狼算法(GWO)、鲸鱼优化算法(WOA)和粒子群算法(PSO)有着更好的精度与收敛速度;在二维Otsu图像阈值分割模型上, β -PAVOA在搜到最优解的情况下收敛速度也仍然领先,这也证明了改进算法的有效性。

关键词:元启发式算法;改进非洲秃鹫优化算法; β 分布;分段线性混沌映射;图像阈值分割

中图分类号:TP311 **文献标志码:**A

Research on Image Threshold Segmentation Based on Improved African Vulture Optimization Algorithm

ZHANG Xiaoyu, FANG Zhongqing, DU Yi, KONG Weibin, WANG Yuting, CHENG Ziyao

(School of Information Engineering, Yancheng Institute of Technology, Yancheng 224051, China)

✉ xiaoyuzhang721@163.com; fangzq@ycit.edu.cn; Lemond213@163.com; kongweibin@ycit.cn; 2251193251@qq.com; 3417747952@qq.com

Abstract: Aiming at the shortcomings of the African Vulture Optimization Algorithm (AVOA) in terms of insufficient global search capability and redundant local search strategies, this paper proposes an improved African Vulture Optimization Algorithm (β -PAVOA). Piecewise Linear Chaotic Map (PWLCM) is used to initialize population and enhance population diversity. The β -distribution and hunger-rate-based search strategy are introduced to enhance the global search capability of the algorithm. The local search strategy of the original algorithm is improved to help the algorithm jump out of the local optimum in time. The effectiveness of the algorithm is verified by eight test functions and it is applied to the two-dimensional Otsu image threshold segmentation model. The experimental results indicate that β -PAVOA has better accuracy and convergence speed on the test functions, compared to the African Vulture Algorithm (AVOA), Golden Jackal Optimization (GJO), Grey Wolf Optimization (GWO), Whale Optimization Algorithm (WOA), and Particle Swarm Optimization (PSO). On the two-dimensional Otsu image threshold segmentation model, β -PAVOA still leads in convergence speed even when the optimal solution is found, which also proves the effectiveness of the improved algorithm.

Key words: metaheuristic algorithm; improved African Vulture Optimization Algorithm; β distribution; PWLCM; image threshold segmentation

0 引言(Introduction)

元启发式算法是一类独立于问题的优化算法,能够有效探索一个搜索空间并找到全局最优解。这类算法主要通过模拟

自然界优胜劣汰的规律以及生物行为,实现种群的整体进步,最终求解最优解。典型算法有遗传算法(GA)^[1]、差分进化算法(DE)^[2]、粒子群算法(PSO)^[3]、灰狼算法(GWO)^[4]、鲸鱼优

化算法(WOA)^[5]、金豺狼优化算法(GJO)^[6]等。

按阈值对图像分割是一种常用的图像分割方法,目前国内已经有很多学者将元启发式算法用于图像的阈值分割优化中。KHAIRUZZAMAN等^[7]将灰狼优化算法应用于图像的多级阈值分割,取得了很好的效果。于洋等^[8]为了解决红外相机采集行人图片时图像分割效果问题,提出一种自适应粒子群优化二维 OSTU 的阈值分割算法,能够快速且准确地得到最佳阈值,提高了图像预处理的分割效果。UPADHYAY等^[9]将Kapur熵作为适应度函数,并通过乌鸦搜索算法求得最大Kapur熵,达到最优阈值。

非洲秃鹫算法(AVOA)^[10]是受非洲秃鹫觅食和导航行为启发而提出的高性能智能算法,具有求解精度高的优点,但其全局搜索能力弱、易陷入局部。本文通过分段线性混沌映射(PWLCM)进行种群初始化,增强种群多样性;在全局搜索阶段引入 β 分布更好地控制种群在搜索空间上的重新划分;提出了新的基于饥饿率的全局搜索策略,用于增强算法搜索能力。将改进算法运用在二维 Otsu 图像阈值分割任务中取得了不错的效果。

1 基础非洲秃鹫优化算法 (Basic African Vulture Optimization Algorithm)

非洲秃鹫优化算法(AVOA)是通过非洲秃鹫的觅食行为和生活习惯进行模拟和建模而提出的。在 AVOA 中,非洲秃鹫的生活习惯和觅食行为主要包括以下 4 个阶段。

1.1 阶段一:随机选择最佳秃鹫

初始化种群后首先计算种群的适应度,其次选择最优个体为最优秃鹫,选择次优个体作为次优秃鹫。每次的最佳秃鹫会在这两个个体中通过如下公式计算产生,同时在每次更新迭代后重新计算整个总体。

$$R(i) = \begin{cases} \text{BestVulture}_1, p_i = L_1 \\ \text{BestVulture}_2, p_i = L_2 \end{cases} \quad (1)$$

其中, L_1 和 L_2 为搜索操作之前给定的参数,其值介于 0~1 且两个参数之和为 1。选择最优解的概率 p_i 使用如下公式的轮盘赌方式获得,并为每组选择最优解。

$$p_i = \frac{F_i}{\sum_{i=1}^n F_i} \quad (2)$$

1.2 阶段二:计算秃鹫的饥饿率

秃鹫的饥饿率由如下公式计算得出:

$$t = h \times \left[\sin^{\omega} \left(\frac{\pi}{2} \times \frac{l}{\text{maxiterations}} \right) + \cos \left(\frac{\pi}{2} \times \frac{l}{\text{maxiterations}} \right) - 1 \right] \quad (3)$$

$$F = (2 \times \text{rand}_1 + 1) \times z \times \left(1 - \frac{l}{\text{maxiterations}} \right) + t \quad (4)$$

其中, F 表示秃鹫的饥饿率, l 表示当前的迭代次数, maxiterations 表示最大迭代次数, z 为介于 -1~1 且每代变化的随机数, h 是介于 -2~2 的随机数, rand_1 是介于 0~1 的随机数。

同时,当 F 的绝对值大于 1 时,算法进入探索阶段;当 F 的值小于或等于 1 时,算法进入开发阶段。

1.3 阶段三:探索

AVOA 的探索阶段秃鹫有两种不同的搜索策略,在两组最佳秃鹫之一的周围区域寻找食物和根据其饥饿程度在环境中随机搜索。

$$P(i+1) = R(i) - D(i) \times F \quad (5)$$

$$D(i) = |X \times R(i) - P(i)| \quad (6)$$

公式(5)模拟了在两组最佳秃鹫之一的周围区域寻找食物,其中 $P(i+1)$ 为下一次迭代的位置, F 是由公式(4)计算得出的秃鹫的饥饿率, $R(i)$ 是由公式(1)给出的最佳秃鹫, $D(i)$ 为到两组最佳秃鹫之一的随机距离,通过公式(6)计算得到。在公式(6)中, $R(i)$ 是由公式(1)给出的最佳秃鹫。 X 被用作增加随机运动的系数向量,由 $X = 2 \times \text{rand}$ 计算得到,其中 rand 为介于 0~1 的随机数。 $P(i)$ 为当前秃鹫的位置。

$$P(i+1) = R(i) - F + \text{rand}_2 \times ((ub - lb) \times \text{rand}_3 + lb) \quad (7)$$

公式(7)模拟了秃鹫根据其饥饿程度在环境中随机搜索,其中 rand_2 是介于 0~1 的随机值, lb 和 ub 表示变量的上界和下界, rand_3 是用于增加随机性的系数。

这两种策略会根据 P_1 的值进行选择, P_1 是介于 0~1 的数。设 rand_{P_1} 为介于 0~1 的随机数。如果 rand_{P_1} 的值小于或等于 P_1 ,则使用公式(5),反之,使用公式(7)。

1.4 阶段四:开发

1.4.1 开发(第一阶段)

当 F 的绝对值介于 0.5~1 时,AVOA 进入开发阶段的第一阶段。在开发阶段的第一阶段,执行缓慢围攻和旋转飞行两种不同的策略。

(1)缓慢围攻:当 $|F| \geq 0.5$ 时,秃鹫相对能量充足。当许多秃鹫聚集在一个食物源上时,会在食物获取上引起严重的冲突。身体强壮的秃鹫不喜欢分享食物,而弱小的秃鹫试图聚集并引发小冲突以便获取食物,如下两个公式模拟了这两个过程。

$$P(i+1) = D(i) \times (F + \text{rand}_4) - d(t) \quad (8)$$

$$d(t) = R(i) - P(i) \quad (9)$$

公式(8)中, $D(i)$ 为到两组最佳秃鹫之一的随机距离,通过公式(6)计算得出, rand_4 为介于 0~1 的随机值。公式(9)中, $R(i)$ 是在当前迭代中使用公式(1)选择的最佳秃鹫。

(2)旋转飞行:秃鹫经常进行旋转飞行,可用于模拟螺旋运动。所有秃鹫与最佳和次佳秃鹫中的一只建立了一个螺旋方程,公式如下所示:

$$S_1 = R(i) \times \left[\frac{\text{rand}_5 \times P(i)}{2\pi} \right] \times \cos(P(i)) \quad (10)$$

$$S_2 = R(i) \times \left[\frac{\text{rand}_6 \times P(i)}{2\pi} \right] \times \sin(P(i)) \quad (11)$$

$$P(i+1) = R(i) - (S_1 + S_2) \quad (12)$$

其中, rand_5 和 rand_6 是介于 0~1 的随机数,最后通过公式(12)更新秃鹫的位置。

这两种策略会根据 P_2 的值进行选择, P_2 是介于 0~1 的数。设 rand_{P_2} 为介于 0~1 的随机数。如果 rand_{P_2} 的值小于或等于 P_2 ,则根据公式(8)实施缓慢围攻策略,反之,使用公式(12)执行旋转飞行策略。

1.4.2 开发(第二阶段)

当 $|F| < 0.5$ 时,AVOA进入开发阶段的第二阶段,秃鹫开始进行聚集围攻和争夺食物。

(1)几种秃鹫在食物源上的聚集。

描述秃鹫这种运动的公式如下所示:

$$A_1 = \text{BestVulture}_1(i) - \frac{\text{BestVulture}_1(i) \times P(i)}{\text{BestVulture}_1(i) - P^2(i)} \times F \quad (13)$$

$$A_2 = \text{BestVulture}_2(i) - \frac{\text{BestVulture}_2(i) \times P(i)}{\text{BestVulture}_2(i) - P^2(i)} \times F \quad (14)$$

$$P(i+1) = \frac{A_1 + A_2}{2} \quad (15)$$

公式(13)和公式(14)中, $\text{BestVulture}_1(i)$ 是当前迭代中第一组的最佳秃鹫, $\text{BestVulture}_2(i)$ 是当前迭代中第二组的最佳秃鹫。用公式(15)计算得到下一代秃鹫的位置 $P(i+1)$ 。

(2)对食物的激烈竞争。

当 $|F| < 0.5$ 时,领头的秃鹫变得饥饿和虚弱,没有足够的能量对抗其他秃鹫。而其他秃鹫会从不同方向朝着领头秃鹫的位置移动,可以使用如下公式模拟该现象:

$$P(i+1) = R(i) - |d(t)| \times F \times \text{Levy}(d) \quad (16)$$

其中, $d(t)$ 表示秃鹫与两组中最佳秃鹫之间的距离,该距离通过公式(9)计算得出。莱维飞行机制用于提高公式(16)中AVOA算法的有效性,莱维飞行生成的随机分量用以下公式计算。

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\alpha}}} \quad (17)$$

其中

$$\sigma = \left[\frac{\Gamma(1+\alpha) \times \sin(\frac{\pi\alpha}{2})}{\Gamma(\frac{1+\alpha}{2}) \times \beta \times 2^{\frac{\alpha-1}{2}}} \right]^{\frac{1}{\alpha}} \quad (18)$$

公式(17)、公式(18)中, α 为固定值1.5, u 和 v 服从正态分布:

$$\begin{cases} u \sim N(0, \sigma) \\ v \sim N(0, 1) \end{cases} \quad (19)$$

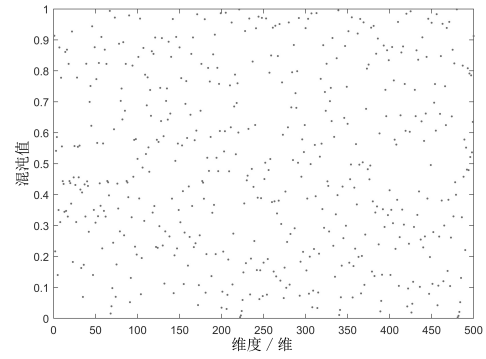
第二阶段的两种策略会根据 P_3 的值进行选择, P_3 是介于0~1的数。设 rand_{P_3} 是介于0~1的随机数,如果 rand_{P_3} 小于或等于 P_3 ,则根据公式(15)更新位置;反之由公式(16)实施对食物的激烈竞争。

2 改进非洲秃鹫优化算法(Improved African Vulture Optimization Algorithm)

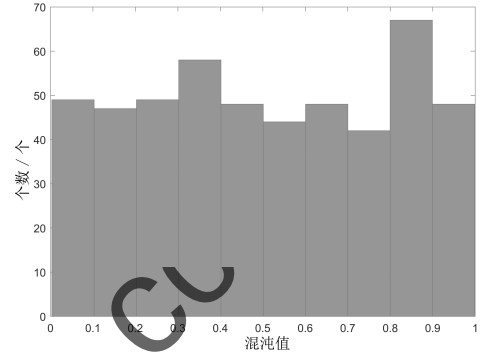
针对非洲秃鹫算法全局搜索能力不足、局部搜索策略冗余及收敛速度慢等缺点, β -PAVOA主要做了以下改进。

2.1 基于PWLCM混沌映射的种群初始化

混沌映射是生成混沌序列的一种方法,被广泛地应用于各种算法中。PWLCM^[11]作为混沌映射的典型代表,其数学形式简单,具有遍历性和随机性。如图1所示,PWLCM在空间中的分布非常均匀。



(a)PWLCM混沌映射图



(b)PWLCM混沌值分布个数

图1 500维下PWLCM混沌映射值分布

Fig.1 Distribution of PWLCM values on 500 dimensions

通过PWLCM初始化种群,可以使 β -PAVOA在初始化种群时更好地搜索整个解空间。PWLCM映射的描述如公式(20)所示:

$$z(t+1) = \begin{cases} \frac{z(t)}{P_z}, & 0 \leq z(t) < P_z \\ \frac{z(t) - P_z}{0.5 - P_z}, & P_z \leq z(t) < 0.5 \\ \frac{1 - P_z - z(t)}{0.5 - P_z}, & 0.5 \leq z(t) < 1 - P_z \\ \frac{1 - z(t)}{P_z}, & 1 - P_z \leq z(t) < 1 \end{cases} \quad (20)$$

公式(20)中, $P_z = 0.4$,如果 $t \geq 2$,则 $z(t)$ 值可根据公式(20)计算得到,如果 $t = 1$,则 $z(1)$ 为映射的初始值, $z(1)$ 取值是介于0~1的随机数。

将生成的值映射到解空间,如下公式所示:

$$P(i) = (ub - lb) \times z(i) + lb \quad (21)$$

其中, $P(i)$ 是映射后的种群位置, ub 和 lb 分别是位置的上界和下界, $z(i)$ 是由公式(20)得到介于0~1的混沌映射随机值。

2.2 基于 β 分布的全局增强搜索策略

β 分布^[12]是指一组定义在区间 $[0, 1]$ 的连续概率分布。在使用元启发式算法解决工程问题时,将 β 分布用于其中很有效,它可以近似模拟包括正态高斯分布在内的几种分布,也允许近似线性或指数递减的分布,可以是一维的,也可以是多维。一维 β 分布表达式如下:

$$\beta(x; p, q, x_0, x_1) = \begin{cases} \left(\frac{x - x_0}{x_c - x_0} \right)^p \left(\frac{x_1 - x}{x_c - x_1} \right)^q, & x \in [x_0, x_1] \\ 0, & \text{其他} \end{cases} \quad (22)$$

$$x_c = \frac{p \times x_1 + q \times x_0}{p + q} \quad (23)$$

其中,通过改变参数 p 和 q 就可以得到可能的 β 分布的多样性。基于 β 分布的全局增强搜索策略如公式(24)所示:

$$P(i+1) = (ub - lb) \times \text{betarand} + lb \quad (24)$$

其中, $P(i+1)$ 为更新后的位置, betarand 是 $p=1, q=1$ 的情况下生成的 β 随机数。

2.3 基于饥饿率 F 的改进搜索策略

在测试过程中发现, AVOA 在探索阶段的全局搜索表现不佳, 因此提出了基于秃鹫饥饿率的改进搜索策略如公式(25)所示, 用于改善这一问题。

$$P(i+1) = (\text{rand} - F) \times X(\text{randi}(1, \text{pop_size})) \quad (25)$$

其中, $P(i+1)$ 为更新后的位置, rand 为介于 $0 \sim 1$ 的随机值。 X 为种群的历史最优位置, $\text{randi}(1, \text{pop_size})$ 为 1 至种群大小之间的随机整数。

2.4 改进局部搜索策略

β -PAVOA 针对局部搜索策略冗余、收敛慢的缺点, 对开发阶段策略进行了精简, 整个开发阶段只使用 AVOA 在开发阶段的第二阶段中的更新策略, rand 是介于 $0 \sim 1$ 的随机数。如果 rand 小于 0.3 , 则根据公式(15)更新位置; 反之由公式(16)实施对食物的激烈竞争。 β -PAVOA 的算法流程如图 2 所示。

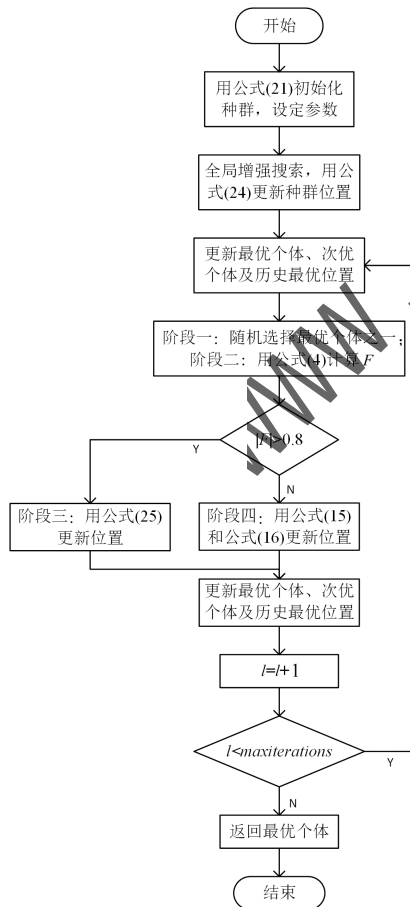


图 2 β -PAVOA 算法流程图

Fig. 2 Flowchart of the β -PAVOA

3 实验仿真与结果分析 (Experimental simulation and result analysis)

3.1 实验环境设置

β -PAVOA 在 8 个基准测试函数上进行了测试, 选取了非洲秃鹫算法 (AVOA)、金豺狼优化算法 (GJO)、灰狼算法 (GWO)、鲸鱼优化算法 (WOA) 和粒子群算法 (PSO) 作为对比算法。测试函数分为两类: F1~F4 为单峰函数, F5~F8 为多峰函数, 测试函数设置见表 1, 算法默认参数设置见表 2。将测试结果进行秩和检验, 确定 β -PAVOA 获得结果的平均值与对比算法之间是否存在显著差异, 当 $p > 0.05$ 时, 认为 β -PAVOA 与对比算法的寻优性能相当。种群大小设置为 50, 最大迭代次数为 1 000 次, 测试维度为 30。为了减少不确定性, 每个测试独立进行 30 次, 计算 30 次测试的平均数和方差。实验是在配备 32 GB RAM 与 i7-12700H 处理器的个人计算机上进行的。

表 1 测试函数设置

Tab.1 Benchmark function settings

编号	函数	维数/维	范围	最小值
F1	Sphere Function	30	$[-100, 100]$	0
F2	Schwefel's Problem 2.22	30	$[-10, 10]$	0
F3	Generalized Rosenbrock's Function	30	$[-30, 30]$	0
F4	Step Function	30	$[-100, 100]$	0
F5	Generalized Schwefel's Problem 2.26	30	$[-500, 500]$	-12 569.487
F6	Generalized Griewank's Function	30	$[-600, 600]$	0
F7	Generalized Penalized Function 1	30	$[-50, 50]$	0
F8	Generalized Penalized Function 2	30	$[-50, 50]$	0

表 2 算法参数设置

Tab.2 Algorithm parameter settings

算法	参数设置
β -PAVOA	$P_z=0.4$; $p=2$; $q=10$; $P_1=0.6$
AVOA	$L_1=0.8$; $L_2=0.2$; $w=2.5$; $P_1=0.6$; $P_2=0.4$; $P_3=0.6$
GJO	$C=1.5$; $\beta=1.5$
GWO	$a=[2, 0]$
WOA	$a=[2, 0]$; $b=1$
PSO	$w=0.3$; $c_1=c_2=1$

3.2 精度与收敛性分析

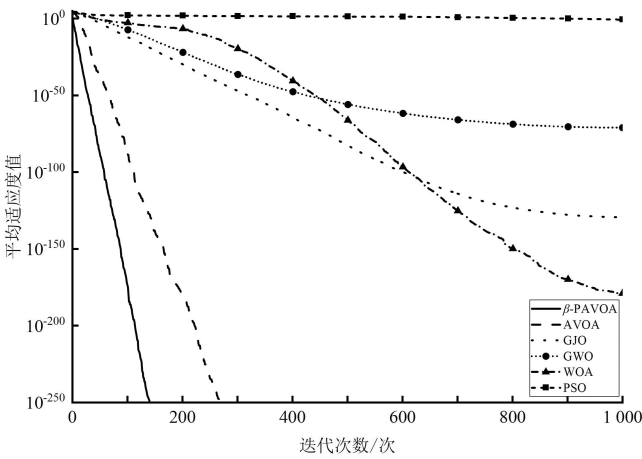
β -PAVOA 与其他算法的测试结果见表 3 和表 4。在 F1~F8 上, β -PAVOA 展现出了极强的优势, 从表 3 和表 4 中可以看出, 相比其他算法在 F3、F4、F7、F8 上的测试结果, β -PAVOA 有着领先几个量级的精度, 在 F1、F2、F5、F6 上的测试结果相近的情况下, 其稳定性有着明显的优势。同时, 从图 3 和图 4 中的算法收敛曲线可以明显看出, β -PAVOA 在 F1~F8 上的收敛精度与收敛速度均优于其他算法。

表3 各类算法在 F1~F4 上的测试结果
Tab.3 Test results of various algorithms on F1~F4

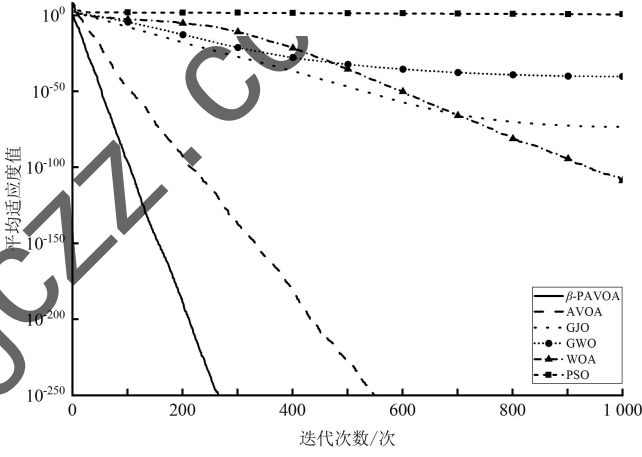
函数	指标	β -PAVOA	AVOA	GJO	GWO	WOA	PSO
F1	Best	0	0	2.16e-131	7.55e-73	4.41e-188	3.89e-2
	Worst	0	0	3.68e-127	2.13e-69	1.20e-174	3.40e-1
	AVE	0	0	2.07e-252	1.69e-70	7.83e-176	1.33e-1
	STD	0	0	6.90e-128	4.14e-70	0	6.57e-2
	ρ	NaN	NaN	1.21e-12	1.21e-12	1.21e-12	1.21e-12
F2	Best	0	0	1.70e-75	3.75e-42	9.67e-118	3.40e-1
	Worst	0	5.75e-318	9.24e-74	2.12e-40	1.50e-107	4.02e+1
	AVE	0	3.96e-319	1.86e-74	4.75e-41	6.88e-109	6.70e+0
	STD	0	0	2.06e-74	4.45e-41	2.79e-108	8.44e+0
	ρ	NaN	1.70e-8	1.21e-12	1.21e-12	1.21e-12	1.21e-12
F3	Best	3.56e-11	2.01e-6	9.61e+1	9.55e-1	9.64e+1	3.57e+4
	Worst	3.86e-5	5.30e-5	9.87e+1	9.84e+1	9.81e+1	1.88e+5
	AVE	9.27e-6	1.90e-5	9.82e+1	9.71e+1	9.72e+1	7.98e+4
	STD	1.10e-5	1.36e-5	6.42e-1	8.76e-1	4.53e-1	3.22e+4
	ρ	NaN	9.52e-4	3.02e-11	3.02e-11	3.02e-11	3.02e-11
F4	Best	1.23e-16	2.39e-11	1.00e+0	1.17e-5	1.32e-3	4.29e-2
	Worst	1.82e-13	5.02e-10	3.00e+0	1.00e+0	2.52e-1	1.67e-1
	AVE	1.98e-14	1.49e-10	2.34e+0	3.35e-1	1.20e-2	4.53e-1
	STD	4.81e-14	1.31e-10	4.76e-1	1.90e-1	4.53e-2	9.56e-2
	ρ	NaN	3.02e-11	3.02e-11	3.02e-11	3.02e-11	3.02e-11

表4 各类算法在 F5~F8 上的测试结果
Tab.4 Test results of various algorithms on F5~F8

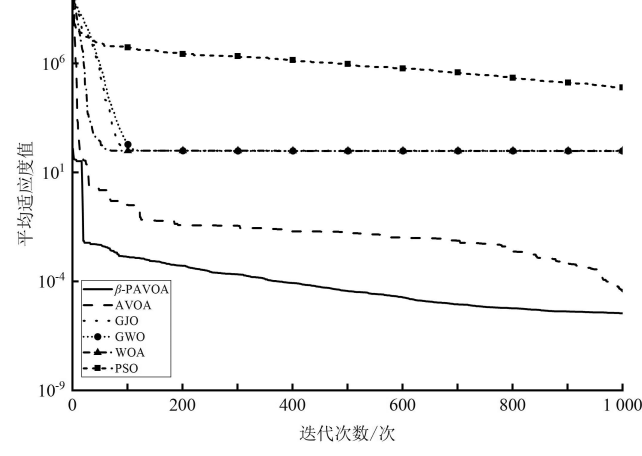
函数	指标	β -PAVOA	AVOA	GJO	GWO	WOA	PSO
F5	Best	-1.26e+4	-1.26e+4	-7.16e+3	-7.37e+3	-1.26e+4	-8.18e+3
	Worst	-1.26e+4	-1.13e+4	-2.87e+3	-5.37e+3	-8.05e+3	-5.21e+3
	AVE	-1.26e+4	-1.25e+4	-4.29e+3	-6.26e+3	-1.16e+4	-6.85e+3
	STD	3.20e-8	2.57e+2	1.27e+3	5.39e+3	1.43e+3	7.90e+2
	ρ	NaN	1	1.86e-1	3.02e-11	3.02e-11	3.02e-11
F6	Best	0	0	0	0	0	7.61e+1
	Worst	0	0	0	5.68e-14	1.13e-13	2.11e+2
	AVE	0	0	0	5.68e-15	3.78e-15	1.43e+2
	STD	0	0	0	1.73e-14	2.07e-14	3.19e+1
	ρ	NaN	NaN	NaN	8.14e-2	3.34e-1	1.21e-12
F7	Best	5.85e-17	3.2e-12	6.65e-2	9.73e-7	1.96e-4	1.04e-4
	Worst	6.65e-13	4.65e-11	6.22e-1	7.23e-2	9.53e-3	9.37e-3
	AVE	6.21e-14	1.18e-11	1.96e-1	2.90e-2	1.14e-3	1.69e-3
	STD	1.31e-13	8.71e-12	1.03e-1	1.51e-2	1.81e-3	2.07e-3
	ρ	NaN	3.01e-11	3.01e-11	3.01e-11	3.01e-11	3.01e-11
F8	Best	2.86e-16	9.49e-11	1.00e+0	1.7e-5	2.20e-3	4.90e-3
	Worst	3.44e-12	1.33e-9	1.91e+0	7.55e-1	2.16e-1	1.57e-1
	AVE	4.09e-13	5.34e-10	1.48e+0	3.21e-1	5.15e-2	6.03e-2
	STD	7.18e-13	3.55e-10	1.94e-1	1.59e-1	6.03e-2	3.76e-2
	ρ	NaN	3.01e-11	3.02e-11	3.02e-11	3.02e-11	3.02e-11



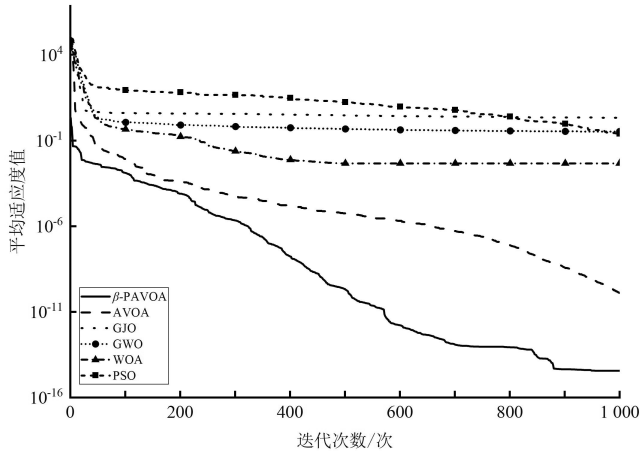
(a) F1 函数的收敛对比曲线



(b) F2 函数的收敛对比曲线



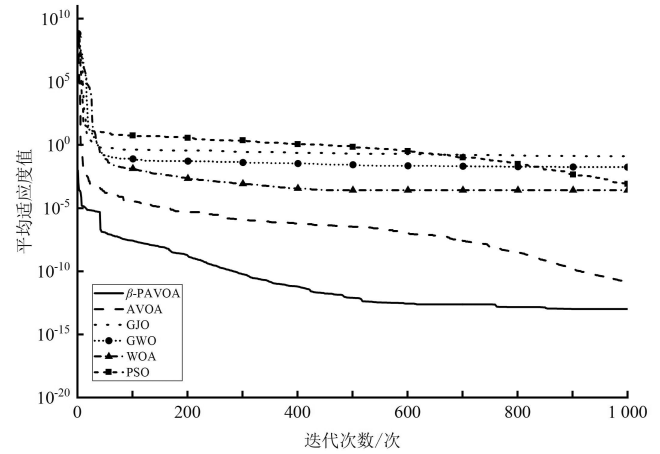
(c) F3 函数的收敛对比曲线



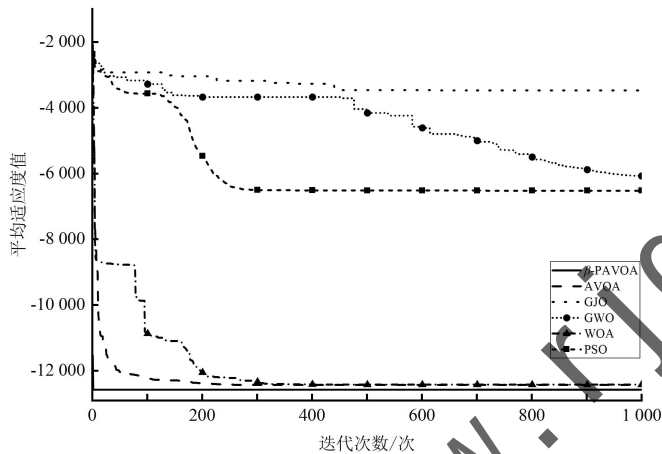
(d) F4 函数的收敛对比曲线

图 3 不同算法的 F1~F4 收敛曲线对比

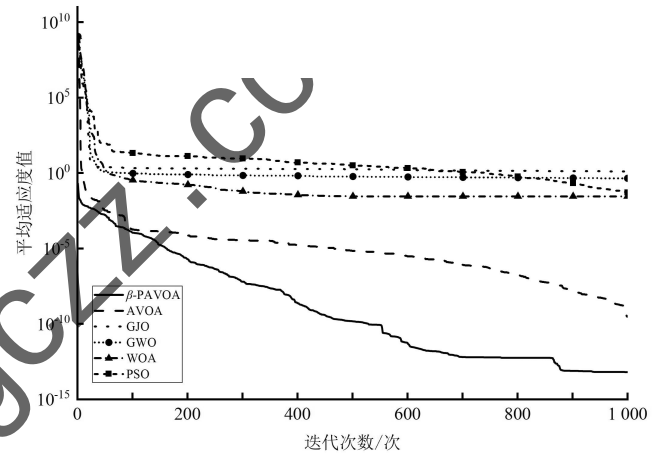
Fig. 3 Comparison of convergence curves of different algorithms on F1~F4



(c) F7 函数的收敛对比曲线



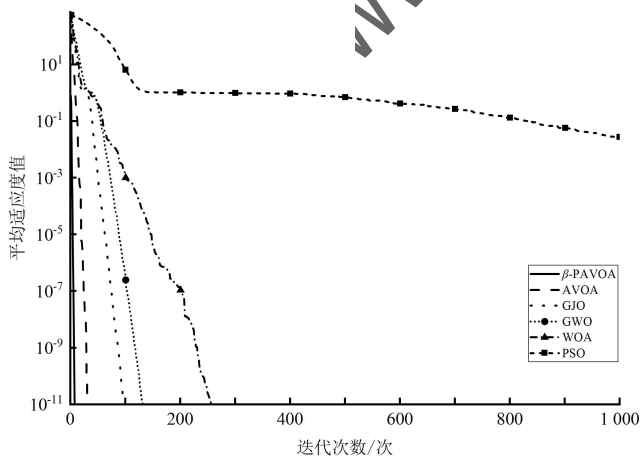
(a) F5 函数的收敛对比曲线



(d) F8 函数的收敛对比曲线

图 4 不同算法的 F5~F8 收敛曲线对比

Fig. 4 Comparison of convergence curves of different algorithms on F5~F8



(b) F6 函数的收敛对比曲线

4 Otsu 图像阈值分割 (Otsu image threshold segmentation)

Otsu 阈值分割常用于图像二值化或阈值化^[13]。图像阈值化是将灰度图像转换为二值图像的过程,其中每个像素根据特定的阈值被赋予黑色或白色值。Otsu 阈值分割法简单高效,适用于分割、目标检测和特征提取^[14]。Otsu 阈值分割法在具有双峰直方图的图像上效果最好,即图像中有清晰的表示背景和前景像素强度的峰值。对于具有更复杂分布的图像,其他阈值化方法或更高级的技术可能更合适。

Otsu 阈值分割法是找到能够使类间方差最大化的阈值,从而有效地将像素分为两个类别:前景和背景。类间方差测量两个类之间的扩展或可分性,并根据图像中的像素强度和概率计算。

4.1 Otsu 阈值分割

假设图形大小为 $M \times N$, 图像灰度范围为 $[0, L-1]$, n_i 为图像灰度级 i 的像素点数, 则灰度级 i 出现的概率为 $p_i = n_i / (M \times N)$ 。在单阈值分割模型中, 图像被分割成两类, 其中 C_0

类像素点灰度级为 $[0, T]$, C_1 类像素点灰度级为 $[T+1, L-1]$ 。 $P_0(T)$ 、 $P_1(T)$ 分别为 C_0 、 C_1 出现的概率, $u_0(T)$ 、 $u_1(T)$ 分别为 C_0 、 C_1 的平均灰度级, 由如下公式计算得出:

$$P_0(T) = \sum_{i=0}^T p_i \quad (26)$$

$$P_1(T) = \sum_{i=T+1}^{L-1} p_i = 1 - P_0(T) \quad (27)$$

$$u_0(T) = \sum_{i=0}^T \left[i \frac{p_i}{P_0(T)} \right] \quad (28)$$

$$u_1(T) = \sum_{i=T+1}^{L-1} \left[i \frac{p_i}{P_1(T)} \right] \quad (29)$$

$$u = \sum_{i=1}^L i p_i = P_0(T) \times u_0(T) + P_1(T) \times u_1(T) \quad (30)$$

$$\delta_b^2(T) = P_0(T) \times [u_0(T) - u]^2 + P_1(T) \times [u_1(T) - u]^2 \quad (31)$$

$$T^* = \operatorname{argmin}_{(1 \leq T < L)} \{ \delta_b^2(T) \} \quad (32)$$

图像的平均灰度级 u 由公式(30)计算得到, 图像类间方差 $\delta_b^2(T)$ 由公式(31)计算得到。当类间方差达到最大时, 该灰度级 T^* 为最优分割阈值, 即 Otsu 阈值。

4.2 Otsu 多阈值分割

假设 $n-1$ 个阈值 $T_1, T_2, \dots, T_{n-1}$ 将图像分为 n 类, 用 $C_0 = \{0, 1, 2, \dots, T_1\}, \dots, C_n = \{T_{n+1}, T_{n+2}, \dots, L-1\}$ 表示, 则各类出现的概率分别表示为 $P_0, P_1, \dots, P_{n-1}$, 平均灰度级为 $u_0, u_1, \dots, u_{n-1}$, 类间方差分别为 $\delta_0^2(T), \delta_1^2(T), \dots, \delta_{n-1}^2(T)$, 其运算公式如下所示:

$$P_k = \sum_{i=T_k}^{T_{k+1}-1} p_i \quad (33)$$

$$u_k = \frac{1}{P_k} \sum_{i=T_k}^{T_{k+1}-1} i p_i \quad (34)$$

$$\delta_k^2 = \sum_{i=T_k}^{T_{k+1}-1} (i - u_k)^2 \frac{p_i}{P_k} \quad (35)$$

$$\delta_b^2 = \sum_{i=0}^{n-1} P_i \delta_i^2 \quad (36)$$

$$\{T_1^*, T_2^*, \dots, T_{n-1}^*\} = \operatorname{argmax}_{(1 \leq T < L)} \{ \delta_b^2 \} \quad (37)$$

图像类间方差用公式(36)表示, 公式(37)则是最终的目标函数。

4.3 二维 Otsu 模型测试结果及分析

在二维 Otsu 模型测试中测试了 β -PAVOA 及对比算法的性能表现, 测试选取了 2 张常用测试图片与 1 张内容复杂的图片(如图 5 所示)。测试结果如图 6 所示, β -PAVOA 在找到最优解时, 收敛速度与稳定性相比其他算法有明显的优势, 这也验证了优化策略的有效性。



(a) 相机



(b) 人像



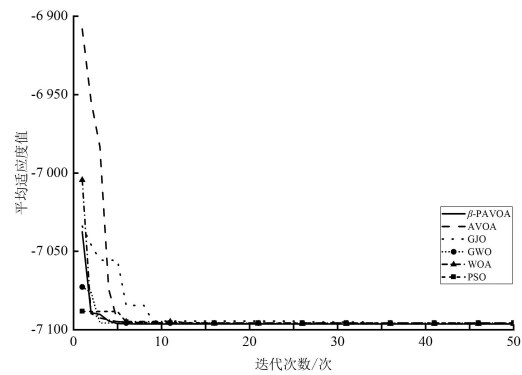
(c) 月球

图 5 测试用图

Fig. 5 The test pictures



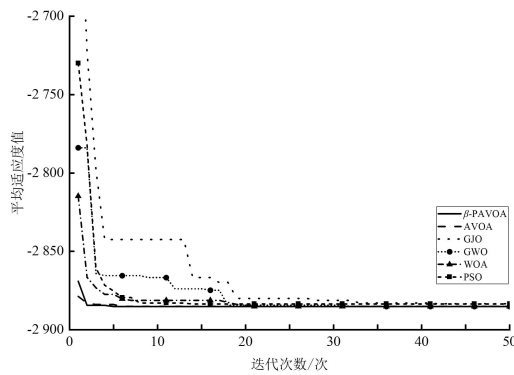
(a) 相机二维 Otsu 测试结果



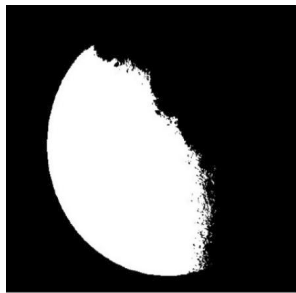
(b) 相机二维 Otsu 模型迭代曲线



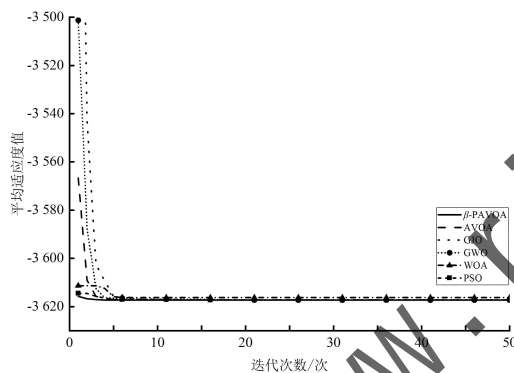
(c) 人像二维 Otsu 测试结果



(d)人像二维 Otsu 模型迭代曲线



(e)月球二维 Otsu 测试结果



(f)月球二维 Otsu 模型迭代曲线

图 6 测试结果图

Fig. 6 The test results graph

5 结论 (Conclusion)

本文提出了一种基于 PWLCM 与 β 分布的非洲秃鹫算法 β -PAVOA。首先,通过 PWLCM 扩大种群在解空间的覆盖程度,提高寻优能力。其次,使用基于 β 分布的全局增强搜索策略增强算法的全局搜索能力。基于饥饿率 F 的改进搜索策略进一步增强了全局搜索能力。最后,通过简化局部搜索策略提升局部收敛速度。为了验证 β -PAVOA 的有效性,通过 8 个基准测试函数和二维 Otsu 图像阈值分割模型与非洲秃鹫算法 (AVOA)、金豺狼优化算法 (GJO)、灰狼算法 (GWO)、鲸鱼优化算法 (WOA) 和粒子群算法 (PSO) 进行对比实验,采用均值、标准差与秩和检验方法进行评估。结果表明, β -PAVOA 算法在探索能力和求解精度上都有显著优势。

参考文献 (References)

[1] HOLLAND J H. Genetic algorithms[J]. Scientific American, 1992, 267(1): 66-73.

- [2] PRICE K V. Differential evolution [M]//ZELINKA I, SNÁŠEL V, ABRAHAM A. Handbook of optimization: from classical to modern approach. Berlin, Heidelberg: Springer, 2013: 187-214.
- [3] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization: an overview[J]. Swarm Intelligence, 2007, 1: 33-57.
- [4] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [5] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [6] CHOPRA N, MOHSIN ANSARI M. Golden jackal optimization: a novel nature-inspired optimizer for engineering applications[J]. Expert Systems with Applications, 2022, 198: 116924.
- [7] KHAIRUZZAMAN A K M, CHAUDHURY S. Multilevel thresholding using grey wolf optimizer for image segmentation[J]. Expert Systems with Applications, 2017, 86: 64-76.
- [8] 于洋, 孔琳, 虞闯. 自适应粒子群集优化二维 OSTU 的图像阈值分割算法[J]. 电子测量与仪器学报, 2017, 31(6): 827-832.
- [9] UPADHYAY P, CHHABRA J K. Kapur's entropy based optimal multilevel image segmentation using crow search algorithm[J]. Applied Soft Computing, 2020, 97: 105522.
- [10] ABDOLLAHZADEH B, GHAREHCHOPOGH F S, MIRJALILI S. African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems[J]. Computers & Industrial Engineering, 2021, 158: 107408.
- [11] 肖玲, 方照, 周扬. 基于双重混沌加密的 OFDM 系统设计与实现[J]. 北京电子科技学院学报, 2021, 29(4): 38-44.
- [12] 李学京. 置信分布的贝塔分布近似及其在可靠性统计中的应用[J]. 强度与环境, 2007, 34(2): 17-23.
- [13] OTSU N. A threshold selection method from gray-level histograms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66.
- [14] 袁小翠, 黄志开, 马永力, 等. Otsu 阈值分割法特点及其应用分析[J]. 南昌工程学院学报, 2019 (1): 85-90.

作者简介:

张啸宇(1996-), 男, 硕士生。研究领域: 信号处理, 智能检测。
方忠庆(1990-), 男, 博士, 讲师。研究领域: 信号处理, 固体激光器。本文通信作者。
杜义(2000-), 男, 硕士生。研究领域: 智能检测。
孔维宾(1982-), 男, 博士, 副教授。研究领域: 信号处理, 微波技术。

王玉婷(2003-), 女, 本科生。研究领域: 电子信息。
程子耀(2004-), 男, 本科生。研究领域: 电子信息。