

文章编号: 2095-2163(2020)08-0071-06

中图分类号: TP391.41

文献标志码: A

基于改进蝗虫优化算法的最大2维熵图像分割

周 娇¹, 王 力^{1,2}, 陈小青¹

(1 贵州大学 大数据与信息工程学院信息与通信工程系, 贵阳 550025; 2 贵州工程应用技术学院 信息工程学院, 贵州 毕节 551700)

摘要: 本文针对图像分割中阈值选取问题, 将改进的蝗虫优化算法应用于图像分割阈值选取中, 不仅减少了计算复杂度, 并且提升了分割精确度。利用贝塔函数产生初始种群, 在初始化过程中种群均匀性及多样性得以保证。引入两部分非线性惯性权重, 第一部分促使种群在全局大范围精确地搜索, 以便得到更为精确的解; 第二部分主要应用于后期加快收敛, 使算法快速跳出局部最优。利用经典基准函数进行了试验仿真验证, 改进后的算法求解质量及收敛速度有较好的表现。

关键词: 图像处理; 蝗虫优化算法; 最大2维熵; 贝塔函数; 非线性惯性权重

Image segmentation of 2-D maximum entropy based on the improved grasshopper optimization algorithm

ZHOU Jiao¹, WANG Li^{1,2}, CHEN Xiaoqing¹

(1 Department of Information and Communication Engineering, College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China; 2 School of Information Engineering, Guizhou University of Engineering Science, Bijie 551700, Guizhou, China)

[Abstract] Aiming at the problem of threshold selection in image segmentation, an improved grasshopper optimization algorithm is proposed for image segmentation threshold selection. The computational complexity is reduced and segmentation accuracy increased. Considering that the original grasshopper optimization algorithm is prone to fall into the disadvantages of local optimization, slow convergence and low optimization accuracy, the initial population is generated by using beta distribution, and the population uniformity and diversity are guaranteed during initialization. Then two parts of nonlinear are introduced. The simulation results show that the improved algorithm has good performance in solving quality and convergence speed.

[Key words] image processing; the grasshopper optimization algorithm; 2-D maximum entropy; beta function; nonlinear inertial weight

0 引言

将最大化或最小化作为待解决函数, 求解待解决函数变量最优值过程叫做优化, 优化问题可应用于各个方面。优化问题可划分为一个目标或若干目标, 按照明确的参数类别、束缚条件和目标数目, 适当优化算法处理问题。通过数学优化手段易落入局部最佳值, 不能得到全局最佳值; 针对未解或大计算量的函数, 其表现没有明显作用。在随机优化方法中, 基于自然的和基于种群的算法最受欢迎。基于种群的算法是通过达尔文演化论获取。如 PSO (particle swarm optimization), 该算法模拟了鸟类或鱼类群的觅食; ACO 算法模仿了蚂蚁从巢穴寻找到食物来源的最短路径的集体行为。

图像分割被应用于医学图像、检测工业缺陷和合成孔径雷达图像等各个领域^[1-2]。通过阈值将图

像分割常见方法有很多, 如最大类间阈值法、最小错误法以及鉴于信息熵法^[3]。于洋等人^[4]提议的二维最大类间阈值法, 应用灰度级和领域灰度制作二元组, 通过 2D-Ostu 模型分割图像, 结果良好; Horng^[5]提出基于 shannon 熵分割图像; 张金果等人^[6]运用最小交叉 tsallis 熵将其改进, 分割效果在噪声抵抗性能上表现较好; 吴诗娅等人^[7]采用了二维 sallis 交叉熵, 将其分解为不是对称的, 降低了分割计算复杂度; 马英辉等^[8]提出 2D_renyi 对图像进行分割, 降低计算复杂度, 缩减了计算时间; 聂方彦^[9]等提出了 tsallis 相对熵单阈值对图像进行分割。图像阈值分割计算量大, 有关这个缺点大多学者通过将群智能算法优点结合于分割算法中。如 PSO (particle swarm optimization)^[10]、ABC (artificial bee colony algorithm)^[11]、GA (genetic algorithm)^[12]

基金项目: 国家新工科实践资助(黔教高函[2018]209号); 贵州省教育厅创新群体重大研究资助(黔教合 KY 字[2016]057)。

作者简介: 周 娇(1991-), 女, 硕士研究生, 主要研究方向: 图像处理、智能算法、机器学习; 王 力(1971-), 男, 学士, 教授, 主要研究方向: 图像处理、机器学习; 陈小青(1992-), 女, 硕士研究生, 主要研究方向: 图像处理、机器学习。

通讯作者: 陈小青 Email: 85825615@qq.com

收稿日期: 2020-06-25

和FA(firefly algorithm)^[13]。但以上算法常常会落入最佳值的局部。2017年, SHAHRZAD SAREMI等人^[14]受蝗虫群行为的启示,创作出蝗虫优化算法(Grasshopper Optimisation Algorithm, GOA),因其具有可调参数少、可操作性强、迅速收敛等优点,广泛应用于各种优化问题中,为二维最大熵分割方法选取最优阈值提供一种新的手段。为了图像分割精度有所提高,本文将改进的蝗虫优化算法用于寻求最优分割阈值,定义基准函数为最大熵函数,通过实验测试分割结果。结果表明,本文中改进的算法最优阈值能准确找到,图像分割实时性的要求能满足。

1 原始蝗虫优化算法

蝗虫是一种有害生物。蝗虫群在它们幼虫期或成年期都存在蜂群行为。幼虫期在短距离内缓慢行动,而在长距离快速行动是成年期的主要表现。蝗虫群行为的数学模型如式(1):

$$X_i = S_i + G_i + A_i. \quad (1)$$

其中, X_i 定义了第 i 个蝗虫的位置; S_i 是社交互动; G_i 是第 i 个蝗虫的重力; A_i 表示风平流。为表示随机行为公式(1)可写为公式(2)。

$$X_i = r_1 S_i + r_2 G_i + r_3 A_i, \quad (2)$$

其中, r_1 、 r_2 和 r_3 表示 $[0, 1]$ 随机数; $G_i = -g \overline{e_g}$; $A_i = -g \overline{e_w}$ 。

$$S_i = \sum_N s(d_{ij}) \overline{d_{ij}}. \quad (3)$$

其中, d_{ij} 表示第 i 个和第 j 个蝗虫之间的距离 $d_{ij} = |x_j - x_i|$; s 是定义强度的函数。 $\overline{d_{ij}} = \frac{x_j - x_i}{d_{ij}}$ 第 i 个和第 j 个蝗虫之间单位矢量; $\overline{e_g}$ 和 $\overline{e_w}$ 分别为朝向地球中心的单位矢量和风向的单位矢量。

定义社会力量的 s 函数计算如式(4)所示:

$$s(r) = f e^{\frac{r}{l}} - e^{-r}. \quad (4)$$

其中, f 表示吸引力的强度; l 表示吸引力的长度尺度。由以上公式可将公式(1)扩展为式(5)。其中, N 为蝗虫的数量。

$$X_i = \sum_{j=1}^N s(|x_j - x_i|) \frac{x_j - x_i}{d_{ij}} - g \overline{e_g} + u \overline{e_w}. \quad (5)$$

然而,由于蝗虫可飞快地抵达安逸范围,同时个体不可能收敛到特指点,因此公式(5)不可直接地应用到处理优化方面。为处理优化问题,则将其修改为式(6)。

$$X_i^d = \sum_{j=1}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j^d - x_i^d}{d_{ij}} + \overline{T}_d. \quad (6)$$

其中, ub_d 是 d 维的上限; lb_d 是 d 维的下限。 \overline{T}_d 是到目前为止找到的最佳解,而 c 是减小舒适区、排斥区和吸引区的递减系数。

式(6)前半部分考虑到当前蝗虫相对其它蝗虫的位置,蝗虫的下一个位置是由所有蝗虫位置决定的。实际上此算法考虑到所有蝗虫状态,而不像粒子群优化算法(Particle Swarm Optimization, PSO)仅考虑当前位置、个体最佳及全局最佳来更新粒子的位置,即除了有利于更新位置的粒子有位置外,而其它粒子则被抛弃。而蝗虫优化算法要求所有蝗虫个体参与,并不存在舍弃。减小舒适区、排斥区和吸引区的递减系数 c 可由式(7)得出。

$$c = c_{max} - l \frac{c_{max} - c_{min}}{L}. \quad (7)$$

值得注意的是,内部 c 有助于减少与蝗虫之间的排斥力,与迭代次数成正比;而外部 c 随着迭代次数增加而减小目标周围的搜索范围。式(7)中 $c_{max} = 1$, 代表取最大数; $c_{min} = 0.000\ 01$, 代表取最小数; l 代表目前迭代次数, L 代表最大迭代数值。

蝗虫算法伪代码如下:

种群个体初始化 $X_i (i = 1, 2, \dots, n)$

初始化参数 (c_{max} 、 c_{min} 、最大迭代次数)

计算每个搜索代理适应度值

T = 最佳搜索代理个体

While ($l <$ 最大迭代次数)

使用公式(7)更新参数 c

对每个搜索代理在区间 $[1, 4]$ 归一化蝗虫间距离

若超过当前上下界返回目前搜索代理

结束 for 循环

若有最佳界,更新 T

$L = l + 1$ (L 和 l 在公式(7)中均有说明)

结束 while 循环

返回 T

2 改进蝗虫优化算法

2.1 贝塔函数初始化

群智能算法通常先初始化种群,为后期计算阶段提供初始种群个体。因此,初始化方式影响算法寻求全局最优的进程(如收敛加快,解的精确度的提高等)。原蝗虫优化算法的初始种群是通过随机方式产生,最佳蝗虫个体在初始种群选取,然后再进行更新迭代。初始种群应尽可能在解空间分布均匀,而通过随机方式产生使得分布不均,造成初始个体位置易陷入局部解附近,算法易在局部寻求最优。因初始化结果优劣决定下一环节,所以本文中不再

采用原算法的方式来随机生成种群, 而是采用贝塔函数来生成。贝塔分布^[15-16] (Beta Distribution) 的密度函数是伯努利分布和二项式分布的共轭先验分布, 是定义在区间 $[0, 1]$ 上连续的概率分布, 且仅有 α 和 β 两个参数, 称为形状参数。当 $\alpha = \beta = 1$ 时, 在区间 $[0, 1]$ 为均匀分布。因可调参数简单, 能在解空间产生均匀分布, 采用其产生初始种群。贝塔函数 (Beta function) 也称 B 函数和第一欧拉积分, 定义如式(8)^[17]:

$$B(P, Q) = \int_0^1 x^{p-1} (1-x)^{q-1} dx. \quad (8)$$

其中, $P > 0, Q > 0$, 贝塔函数满足连续性和对称性。

贝塔函数产生序列, 该序列表示初始种群的可行解, 将可行解进行反向选取。初始化策略步骤为: 首先用贝塔函数产生当前种群的一个可行解 $\{X = (x_1, x_2, \dots, x_d)\}$ (d 为搜索空间的维度; $x_j \in [a_j, b_j]$), 则反向解定义为 $X' = (X_1', X_2', \dots, X_d')$, $x_j = k(a_j + b_j) - x_j$, k 服从 $[0, 1]$ 上的均匀分布。

2.2 非线性惯性权重

为提高原蝗虫优化算法寻优精度和收敛速度, 本文引入非线性惯性权重。由式(6)得到原蝗虫优化算法是通过参数 c 来调整蝗虫间吸引力、排斥和搜索空间大小。有外部系数 c 和内部系数 c , 不管是权衡全局勘探和局部开发的平衡作用, 还是初期勘探与后期开发作用, 它们的共同点都是在迭代过程中, 采取线性递减的策略。前期需要较大 c 值且下降缓慢, 让算法有能力 (即权重大)、有时间 (下降慢) 勘探全局, 以确保解的精确性; 后期权重应当减小且快速下降, 以便算法迅速找到局部最优。而原蝗虫优化算法仅采用同一种策略, 不能满足前期和后期所要的结果, 需将式(7) 更新为式(9):

$$c = w_1 \cdot [c_{max} - l \frac{c_{max} - c_{min}}{L}]. \quad (9)$$

其中, w_1 计算方法为 $w_1 = w_{max} - (w_{max} - w_{min}) * [l/Max_iter]$; $w_{max} = 0.9, w_{min} = 0.4$; 变量 l 是当前迭代次数, Max_iter 为最大迭代次数。则需将式(6) 更新为式(10):

$$X_i^d = \sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s (|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} + w_2 * \overline{T}_d. \quad (10)$$

其中, w_2 计算方法为 $w_2 = w_{max} + (w_{max} - w_{min}) * [(v1/Max_iter) - v]$; $v = (l/Max_iter)^2$ 。基于以上两种手段提出了本文算法 (IGOA)。

2.3 维最大熵图像分割原理

采用图像熵作为分割准则。设原始图像 $f(x, y)$ ($x = 0, 1, \dots, M; y = 0, 1, \dots, N; M \times N$ 为分割图像的大小; $f(x, y) = 0, 1, \dots, L$ 的灰度级为 L ($L = 256$)。以其中一个像素 $f(x, y)$ 及其领域作为一个计算区域, 可得到该像素的灰度均值, 如式(11) 所示。

$$P_{ij} = \frac{n_{ij}}{M \times N} \quad (11)$$

其中, n_{ij} 表示原图点像素点数, P_{ij} 是均值对发生概率。

通过式(11) 可得到如图 1 所示的图像二维直方图。

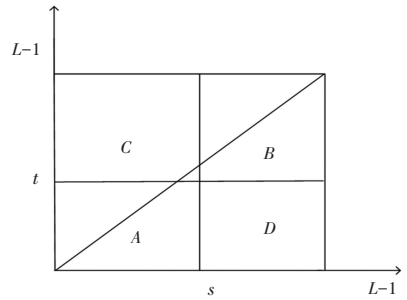


图 1 二维直方图

Fig. 1 2D Histogram

典型情况下, 图像中目标或背景的像素占有最大比例, 所在区域灰度级分布均匀, 且点灰度与区域灰度均值无较大差别。由图 1 可知, 分布在对角线 A 区和 B 区, 反映了图像中目标和背景。点灰度-区域灰度均值对发生的概率主要集中在对角线周围, 即图 1 中 A、B 区。在对角线的 C、D 区, 概率值呈下降趋势, 这两个区反映图像中的边缘点和噪声点等。

综上所述, 为使目标和背景信息熵达到最大, 则应在 A 区和 B 区运用二维最大熵法来确定图像分割的最佳阈值向量 (s, t) 。根据熵的定义, 图像二维熵与阈值向量可通过公式(12) 得到:

$$P_A = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij}, P_B = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} p_{ij}. \quad (12)$$

其中, 二维离散熵的定义为 $H = - \sum_i \sum_j p_{ij} \ln p_{ij}$; 则 A 区和 B 区二维熵为:

$$H(A) = - \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \frac{P_{ij}}{P_A} \ln \frac{P_{ij}}{P_A} = - \frac{1}{P_A} \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij} (\ln p_{ij} - \ln P_A) =$$

$$-\frac{1}{P_A} \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij} \ln p_{ij} + \frac{\ln P_A}{P_A} \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij} = \ln P_A + \frac{H_A}{P_A} \quad (13)$$

同理, $H(B) = \ln P_B + \frac{H_B}{P_B}$ 。其中 $H_A =$

$$-\sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij} \ln p_{ij}, H_B = -\sum_{i=s}^{L-1} \sum_{j=t}^{L-1} p_{ij} \ln p_{ij}。因 C 区和 D 区主要是关于噪声与边界的信息,具有很小的概率 ($p_{ij} = 0$),可忽略不计。则可得到 $P_B = 1 - P_A, H_B = H_L - H_A$ 。其中 $H_L = -\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} \ln p_{ij}, H(B) = \ln(1 - P_A) + \frac{H_L - H_A}{1 - P_A}$ 。$$

则熵的判别函数由式(14)定义:

$$\delta(s, t) = H(A) + H(B) = \ln P_A + \frac{H_A}{P_A} + \ln P_B + \frac{H_B}{P_B} =$$

表1 测试函数

Tab. 1 Test functions

测试函数表达式	维度	范围	最优值
$f_1 = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
$f_9 = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	30	[-5.12, 5.12]	0
$f_{10} = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	[-32, -32]	0
$f_{11} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0

试验结果见表2。

表2 测试结果

Tab. 2 Test function results

函数	算法	平均值	标准差
F1	GOA	119.982 4	148.936 2
	IGOA	1.24e-09	1.86e-09
F2	GOA	2.042 8	3.747 8
	IGOA	1.70e-02	3.03e-01
F9	GOA	20.597 1	9.329 0
	IGOA	1.52e+01	1.32e-01
F10	GOA	40.689 6	0.109 8
	IGOA	1.93e+01	2.20e-01
F11	GOA	0.2055	0.1320
	IGOA	1.35e+01	1.63e+00

函数 f_1 和 f_2 为单峰函数;函数 f_9, f_{10} 和 f_{11} 为多峰函数。单峰函数仅有一个全局最优解而不存在局

$$\ln[P_A(1 - P_A)] + \frac{H_A}{P_A} + \frac{H_L - H_A}{1 - P_A} \quad (14)$$

由以上公式可得熵函数 $(s, t) = H(A) + H(B)$, 所选取的最佳分割阈值 (s^*, t^*) 应满足:

$$(s^*, t^*) = \max\{\delta(s, t)\}。$$

由于每一点灰度-区域灰度均值对的二维熵都要进行计算,计算量呈指数增加,因此有必要引用群智能优化算法。

3 实验仿真与结果分析

3.1 原 GOA 与 IGOA 的对比

本文在同等实验条件下,将原 GOA 与 IGOA 进行对比。其中,算法迭代次数为 500,种群大小为 30。为验证本文算法(IGOA)的鲁棒性及改进点的优化效果,实验在 23 个基准函数中选取 5 个进行测试。测试函数表达式见表 1。

部最优解,是用来测试优化算法的收敛速度。多峰函数存在多个局部最优值,主要测试优化算法跳出局部最优和寻求全局最优的能力。本文将平均值和标准差作为指标,平均值可以反映优化算法的寻优能力,而标准差反映了算法的稳定性。从表 2 可以见,在单峰函数和多峰函数上,虽然没达到理论最优值,但 IGOA 的两个指标都优于 GOA,说明本文增加的改进点是有效的。从图 2 不同算法收敛曲线上可以看出,IGOA 收敛速度和求解精度比 GOA 表现好。

3.2 基于 IGOA 优化图像最大熵

算法 IWOA 阈值优化步骤如下:

(1) 采用贝塔函数初始化 IGOA 参量,包括种群大小 $X_i (i = 1, 2, \dots, 3)$; 初始化 c_{max} 和 c_{min} 和迭代次数 T ;

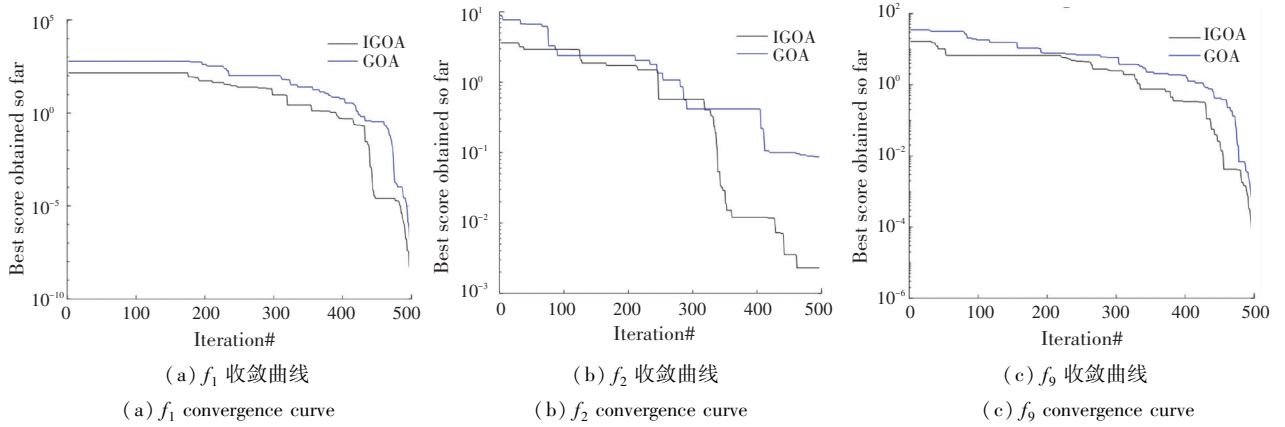


图 2 不同算法收敛速度比较

Fig. 2 Comparison of convergence rate with different algorithm

(2) 利用 IGOA 求分割阈值, 目标函数表达式如式(15)所示:

$$\delta(s, t) = H(A) + H(B) = \ln[P_A(1 - P_A)] + \frac{H_A}{P_A} + \frac{H_L - H_A}{1 - P_A} \quad (15)$$

(3) 基于式(9)和式(10)更新蝗虫个体位置;

(4) 基于式(15)求蝗虫的适应值, 将最小的适应值作为最优值;

(5) 判断是否达到最大迭代次数, 若达到最大迭代次数, 输出最佳阈值 $X^* = (s^*, t^*)$ 结束计算, 否则返回步骤(3)。

为验证本文算法优化最大熵获取最佳分割阈值, 将改进算法与原算法在同一编译语言进行实验对比。种群规模为 20, 迭代次数为 30。本文中在

MATLAB 自带图片数据集 cameraman.tif 和 coins.png 两组图片进行验证。

文中所有对比算法均采用 MATLAB 语言。实验环境为: Window10 系统, 4G 内存, MATLAB R2016b 平台。

将 IGOA 应用于最大二维熵, 从图 3 第一组照片可以看出 IGOA 分割出的图像比 GOA 分割出的图像视觉效果更好, 银币上细节成分也比 GOA 分割出的显示得更多。说明 IGOA 结合二维最大熵, 提取的信息量多。从图 3 第二组相片也可以看出, IGOA 分割出的图像比 GOA 分割出的图像效果更为明显。从表 3 可以看出, 本文算法(IGOA)获得熵值更大, 熵值越大表明分割图像中包含细节越多。从时间上看, IGOA 耗时最少。



(a) 银币原图 (b) 原 GOA 分割图 (c) IGOA 分割图
(a) Original image of silver coin (b) Original GOA segmentation image (c) IGOA segmentation image



(d) 原图 (e) 原 GOA 分割图 (f) IGOA 分割图
(d) Original image (e) Original GOA segmentation image (f) IGOA segmentation image

图 3 不同分割算法分割图比较

Fig. 3 Comparison of segmentation with different algorithm