

文章编号: 1001-3806(2022)02-0274-09

## 基于改进麻雀算法的最大2维熵分割方法

柳长安,冯雪菱,孙长浩,赵丽娟

(华北电力大学 控制与计算机工程学院,北京 102206)

**摘要:** 为了提高最大2维熵分割的性能,提出了基于改进麻雀算法的最大2维熵分割方法,可减小运算量并且缩短计算时间。首先,融合反向学习策略和自适应 $t$ 分布变异,引入精英粒子,以扩大算法搜索范围,增加算法后期局部搜索能力;其次,使用萤火虫机制,对最优解进行扰动变异,进一步增加种群多样性;最后,采用提出的改进麻雀算法寻找图像最大2维熵,得到最优阈值分割图像。结果表明,4幅图像的平均运行时间为0.3695s,远低于基础2维熵算法的1.7547s和基础2维Otsu算法的5.7936s。所提出的改进麻雀算法的全局搜索和局部寻优能力相比原麻雀算法有较大改善,缩短了传统最大2维熵图像分割方法的运行时间,在峰值信噪比和结构相似度指标上均得到提升,具有一定的应用价值。

**关键词:** 图像处理;智能优化算法;麻雀搜索算法;最大2维熵; $t$ 分布

**中图分类号:** TP391.41;TP18 **文献标志码:** A **doi:**10.7510/jgjs.issn.1001-3806.2022.02.020

## Maximum 2-D entropy image segmentation method based on improved sparrow algorithm

LIU Chang'an, FENG Xueling, SUN Changhao, ZHAO Lijuan

(School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China)

**Abstract:** In order to improve the performance of the maximum 2-D segmentation, an image segmentation method based on improved sparrow algorithm (ITSSA) was proposed, which can decrease the amount of computation and shorten the time. Firstly, the reverse learning strategy and adaptive  $t$ -distribution variation were combined, while elite particles were introduced to expand the search range of the algorithm and to increase the local search ability of the algorithm in the later stage. Secondly, the firefly mechanism was used to perturb and mutate the optimal solution for the further increasement of the population diversity. Finally, the improved sparrow algorithm was used to find the maximum 2-D entropy of the image, and then the optimal threshold segmentation image was obtained. The results show that, the average running time of the proposed algorithm in the four images is 0.3695s, which is much lower than 1.7547s of the basic two-dimensional entropy algorithm and 5.7936s of the basic two-dimensional Otsu algorithm. The global search and local optimization ability of ITSSA, compared with the original sparrow algorithm, improves a lot, and the proposed segmentation method in this paper greatly shortens the traditional maximum 2-D entropy image segmentation method of running time. Apart from that, both the peak signal to noise ratio and the feature similarity index of this method increase, which has a certain application value.

**Key words:** image processing; intelligent optimization algorithm; sparrow search algorithm; maximum 2-D entropy; $t$ -distribution

### 引言

图像分割是图像处理的关键步骤<sup>[1]</sup>,分割依据主要包括阈值<sup>[2-3]</sup>、区域<sup>[4-5]</sup>、边缘<sup>[6-7]</sup>。阈值分割原理是根据某种规则求出最优阈值,根据最优阈值进行分割,因其性能相对稳定且实现简单而得到广泛使用。1979

年提出的Otsu法只考虑了灰度信息。后来提出的最大Kapur熵分割法优于Otsu算法,但抗噪性较差<sup>[8]</sup>。除了Kapur熵,一些学者引入交叉熵<sup>[9]</sup>、Renyi熵<sup>[10]</sup>等进行图像分割,虽然提升了图像分割质量,但是1维分割方法性能提升有限。LIU等人<sup>[11]</sup>针对1维分割精度较低的情况提出2维Otsu算法,考虑像素邻域信息,改善了1维Otsu算法性能。ABUTALEB<sup>[12]</sup>提出2维最大熵方法,在分割性能上优于1维最大熵分割方法,但是大大增加运算时间。

群体智能优化算法以其独特的生物群体行为生存

作者简介:柳长安(1971-),男,教授,现主要从事智能机器人、图像处理及电力巡检的研究。

E-mail:liuchangan@ncepu.edu.cn.

收稿日期:2021-01-08;收到修改稿日期:2021-06-04

方式,提供了更多优化问题的新思路。自“群体智能”概念被首次提出,众多学者提取生物种群特征,凝练出新的群智能优化算法。随着近些年群体智能优化算法的蓬勃发展,群体智能算法广泛应用于解决参量最优化问题上,其中包括蚁群算法(ant colony optimization, ACO)<sup>[13]</sup>与粒子群算法(particle swarm algorithm, PSO)<sup>[14]</sup>两大经典优化算法,以及萤火虫算法(firefly algorithm, FA)<sup>[15]</sup>、灰狼算法(grey wolf optimization, GWO)<sup>[16]</sup>、蚁狮算法(ant lion optimizer, ALO)<sup>[17]</sup>、鲸鱼算法(whale optimization algorithm, WOA)<sup>[18]</sup>、正弦余弦算法(sine cosine algorithm, SCA)<sup>[19]</sup>等新型优化算法。

受麻雀在自然中觅食策略的启发,2020年,XUE等人提出麻雀搜索算法(sparrow search algorithm, SSA)<sup>[20]</sup>。SSA算法相比其它算法具有稳定性较高、收敛精度较好和一定程度上避免陷入局部最优等优点,但是SSA算法收敛速度较快容易导致陷入局部最优解的情况,依然有概率得到不可行解。因此,本文中提出了基于精英反向的改进型 $t$ 分布麻雀算法(improved  $t$ -distribution SSA, ITSSA),加强麻雀粒子之间的交流,加快算法收敛速度,同时使用精英反向学习策略保证麻雀种群均匀性并且增强全局搜索的能力,引入 $t$ 分布算法前期增加全局搜索能力,后期增加局部寻优能力,采用萤火虫算法,进一步增加种群多样性。将ITSSA算法应用于最大2维熵分割中,实验表明,此方法可提高图像分割实时性,在峰值信噪比和特征相似程度上得到极大提升。

## 1 麻雀搜索算法

SSA是一种新型的群体智能优化算法,每只麻雀位置对应一个解,更新方式可分为向当前最优位置靠近和向原点靠近。作为群居鸟类的麻雀,与其它鸟类相比,具有聪明且记忆力强的特点。麻雀搜索算法通过麻雀搜索食物和反捕食操作进行参量优化,初始位置用以下矩阵表示:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ \vdots & & \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,j} & \cdots & x_{n,d} \end{bmatrix} \quad (1)$$

式中, $n$ 是麻雀的个数, $d$ 是待优化变量的维度。第 $i$ 只麻雀位置为 $\mathbf{X}_i = [x_{i,1}, \dots, x_{i,j}, \dots, x_{i,d}]$ 。麻雀的适应度表示如下:

$$\mathbf{F} = [f(\mathbf{X}_1) \quad \cdots \quad f(\mathbf{X}_i) \quad \cdots \quad f(\mathbf{X}_n)]^T \quad (2)$$

式中, $f(\mathbf{X}_i)$ 表示麻雀个体的适应度。

麻雀群体分为3种角色:发现者、追随者、警戒者。

发现者通常负责整个麻雀群体食物来源的寻找并且为加入的追随者提供觅食方向。当报警值大于安全值,发现者将追随者带到安全区域。在每次迭代中,发现者的位置更新如下:

$$\mathbf{X}_i^{(R+1)} = \begin{cases} \mathbf{X}_i^{(R)} \cdot \exp\left(\frac{-i}{\alpha_1 \cdot R_{\max}}\right), (R_a < R_s) \\ \mathbf{X}_i^{(R)} + Q \cdot \mathbf{L}, (R_a \geq R_s) \end{cases} \quad (3)$$

式中, $R$ 是当前迭代次数, $R_{\max}$ 是最大迭代次数, $\alpha_1$ 是 $(0,1]$ 范围的随机数, $Q$ 是服从正态分布的随机数, $\mathbf{L}$ 是各元素都为1的1行多维矩阵。 $R_a \in [0,1]$ 和 $R_s \in [0.5,1.0]$ 分别代表预警值和安全阈值。

在每次迭代中,追随者位置更新如下:

$$\mathbf{X}_i^{(R+1)} = \begin{cases} Q \cdot \exp\left[\frac{\mathbf{X}_w^{(R)} - \mathbf{X}_i^{(R)}}{i^2}\right], (i > \frac{n}{2}) \\ \mathbf{X}_b^{(R+1)} + |\mathbf{X}_i^{(R)} - \mathbf{X}_b^{(R+1)}| \cdot \begin{cases} A^T(\mathbf{A}\mathbf{A}^T)^{-1}, (i \leq \frac{n}{2}) \end{cases} \end{cases} \quad (4)$$

式中, $\mathbf{X}_b$ 和 $\mathbf{X}_w$ 分别表示最优位置和最劣位置, $\mathbf{A}$ 是元素为1或-1的1行多维矩阵。

随机产生麻雀群体的警戒者,占整个麻雀群体的10%~20%,其位置更新如下:

$$\mathbf{X}_i^{(R+1)} = \begin{cases} \mathbf{X}_b^{(R)} + \beta_1 \cdot |\mathbf{X}_i^{(R)} - \mathbf{X}_b^{(R)}|, (f_n > f_b) \\ \mathbf{X}_i^{(R)} + \alpha_2 \cdot \frac{|\mathbf{X}_i^{(R)} - \mathbf{X}_b^{(R)}|}{(f_n - f_w) + \beta_2}, (f_n = f_b) \end{cases} \quad (5)$$

式中, $\beta_1$ 是步长控制参量,服从 $(0,1)$ 的正态分布; $\alpha_2$ 是 $[-1,1]$ 范围的随机数; $\beta_2$ 是最小常数,避免当 $f_n = f_b$ 时,分母为0的情况发生。 $f_n, f_b$ 和 $f_w$ 分别代表当前适应度、最佳适应值和最劣适应度。

## 2 融合自适应 $t$ 分布的多策略麻雀搜索算法

### 2.1 自适应 $t$ 分布

$t$ 分布<sup>[21]</sup>又称为学生分布,其曲线形态和自由度参量 $k$ 有关,概率密度函数如下:

$$p(x) = \frac{\Gamma\left(\frac{k+1}{2}\right)}{\sqrt{k\pi}\Gamma\left(\frac{k}{2}\right)} \cdot \left(1 + \frac{x^2}{k}\right)^{-\frac{k+1}{2}} \quad (6)$$

式中, $\Gamma\left(\frac{k+1}{2}\right) = \int_0^{+\infty} x^{\frac{k+1}{2}-1} e^{-x} dx$ 为第2类欧拉积分。

当 $k=1$ 时, $t$ 分布为柯西分布;当 $k \rightarrow \infty$ 时, $t$ 分布为高斯分布。 $t$ 分布融合柯西分布和高斯分布的特点,在算法运行过程中通过改变 $k$ 的值可以获得不同

的变异幅度。在算法运行的初始阶段,  $t$  分布因为  $k$  值较小近似于柯西分布变异,  $t$  分布具有较强的全局探索能力; 在算法运行的中间阶段,  $t$  分布从柯西分布变异向高斯分布变异逐渐过渡, 结合两者优势; 在算法运行后期,  $t$  分布因为  $k$  值较大近似于高斯分布,  $t$  分布具有较强的局部搜索能力, 有利于算法高效找到全局极值点。

本文中采用自适应  $t$  分布变异, 第  $i$  个体原位置  $X_i$  变异公式如下:

$$X_i^T = X_i + \zeta \cdot t(R) \quad (7)$$

式中,  $X_i^T$  代表  $t$  变异后个体位置,  $\zeta$  代表自适应因子,  $t(R)$  为以当前迭代次数  $R$  为参量的  $t$  分布。

## 2.2 精英反向学习策略

反向学习策略 (opposition-based learning, OBL)<sup>[22]</sup> 已被证实可以有效提高智能优化算法搜索能力, 提高近 50% 找到全局最优解的概率, 其主要思想是求问题可行解的反向解, 对原可行解和反向解进行评估, 选出更优解作为下一代个体。精英反向学习 (elite opposition-based learning, EOBL) 利用精英个体包含更多信息的特点, 构造当前种群的反向种群, 选取当前种群和反向种群中更优质的个体作为初始解。设当前群体的精英个体位置为:

$$X_i' = [x_{i,1}', \dots, x_{i,j}', \dots, x_{i,d}'] \quad (8)$$

则精英个体的反向解为:

$$X_i'^* = \alpha_3 [\min(x_{i,j}') + \max(x_{i,j}')] - X_i' \quad (9)$$

式中,  $\alpha_3$  是 (0, 1) 范围的随机数。利用随机生成的  $\alpha_3$  值可以生成多个精英反向解, 如果生成的反向解超出上下边界值, 则重置反向解。

## 2.3 萤火虫算法

萤火虫算法<sup>[15]</sup> 模拟萤火虫发光行为, 亮度弱的萤火虫受亮度强的萤火虫吸引, 进行移动从而更新位置得到新的位置, 亮度强的萤火虫进行扰动操作。萤火虫亮度  $L$  和吸引力  $r$  如下式所示:

$$\begin{cases} L = f(X_i) \exp(-\mu l_{A,B}^2) \\ \gamma = \gamma_{\max} \exp(-\mu l_{A,B}^2) \end{cases} \quad (10)$$

式中,  $f(X_i)$  代表第  $i$  只萤火虫适应度值, 也是其亮度,  $\gamma_{\max}$  代表萤火虫间的最大吸引力,  $\mu$  是光强系数,  $l_{A,B}$  表示两个萤火虫之间的距离。

当萤火虫 A 被萤火虫 B 吸引时, 萤火虫 A 的位置如下式:

$$X_A = X_A + \gamma \times (X_B - X_A) + \lambda(\alpha_4 - 0.5) \quad (11)$$

式中,  $X_A$  和  $X_B$  分别表示萤火虫 A 和萤火虫 B 的位

置。  $\lambda \in [0, 1]$  为步长因子,  $\alpha_4$  为 (0, 1) 服从均匀分布的随机数。

最亮萤火虫 S 的位置更新公式如下:

$$X_S = X_S + \lambda(\alpha_4 - 0.5) \quad (12)$$

式中,  $X_S$  表示最亮萤火虫的位置, (12) 式让萤火虫一定程度上避免过早陷入局部最优。

## 2.4 融合自适应 $t$ 分布的多策略麻雀搜索算法

针对麻雀算法的不足, 首先利用反向学习策略生成反向解加入种群, 扩大算法搜索范围。将迭代次数作为  $t$  分布的自由度参量, 增加种群多样性, 提高算法的全局探索能力和局部开发能力。与此同时, 选择多个最优个体, 利用精英反向学习策略选择发现者, 提高种群质量, 提高算法收敛速度。追随者位置更新易受发现者位置更新情况影响, 为进一步提高收敛精度和寻优效果, 利用萤火虫机制扰动追随者, 改善争夺觅食过程中易陷入全局优化能力。在保留麻雀算法 3 种角色和竞争机制的前提下, 采取动态策略, 设置转换概率  $p$ , 将精英反向策略与萤火虫算法和自适应  $t$  分布变异在一定概率下交替执行, 动态更新位置, 改善基础麻雀算法容易陷入局部最优的缺陷, 达到增强全局寻优能力的目的, 得到更优质的解。具体实现步骤如下。

(1) 初始化参量: 种群数量、最大迭代次数  $R_{\max}$ 、转换概率  $p$ 、发现者比例  $D_p$ 、警戒者比例  $D_s$ 、警戒者预警值  $R_s$  等; (2) 应用第 2.2 节中的反向学习策略生成反向解加入麻雀初始种群, 选取最优个体作为最终的初始种群; (3) 计算每只麻雀适应度并排序; (4) 应用第 2.2 节中的反向学习选择策略选择发现者, 其余作为追随者, 获取精英麻雀动态边界, 更新发现者和追随者的位置, 随机选择侦察警戒者, 更新其位置; (5) 计算并更新每只麻雀适应度值; (6) 判断变异条件: 若随机数  $r \in [0, 1]$  小于转换概率  $p$ , 则利用第 2.1 节中的自适应  $t$  变异公式更新每只麻雀适应度值; 若  $r \geq p$ , 则应用第 2.2 节中的精英反向策略更新发现者, 应用第 2.3 节中的萤火虫算法扰动追随者, 应用原式更新警戒者的适应度值; (7) 对步骤 (6) 的越界个体做越界处理, 比较新适应度值和原值, 若新适应度值优于原值, 则更新位置和适应度值, 反之, 则保留; (8) 判断当前迭代次数  $R$  是否达到最大迭代次数, 若是, 则循环结束, 输出结果, 反之, 返回步骤 (6)。

## 3 基于 ITSSA 的 2 维最大熵图像分割

图像分割的最佳阈值即为彻底分离目标和背景的像素边界。基于最大熵的阈值分割方法就是将熵值作

为目标函数,让图像分割后的目标区域和背景区域熵的和达到最大值。1 维最大熵虽然处理速度较快,但是仅考虑自身像素的灰度信息,而忽略区域相关性,当噪声严重时,因此表现出较差的分割效果以及抗噪性能。2 维最大熵应运而生,结合像素点和区域灰度特征,提取图像有用信息。

2 维最大熵<sup>[23-24]</sup>原理如下:假设输入图像大小为  $M \times N$ ,待分割图像为  $I(x, y)$ ,像素点  $8 \times 8$  邻域灰度均值为  $g(x, y)$ ,且  $1 \leq x \leq M, 1 \leq y \leq N$ 。设  $I(x, y) = u$ ,  $g(x, y) = v$ ,且  $1 \leq u, v \leq L - 1$  (灰度范围  $L = 256$ )。

$$p_{uv} = \frac{q_{uv}}{M \times N} \quad (13)$$

式中,  $q_{uv}$  表示图像值为  $u$  且区域灰度均值为  $v$  的像素点数频率,  $p_{uv}$  是其对应的概率<sup>[20]</sup>。

输入图像 2 维直方图如图 1 所示。

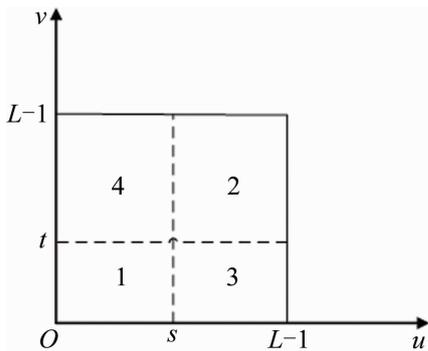


Fig. 1 2-D histogram

区域 1 和区域 2 代表目标和背景,区域 3 和区域 4 代表噪声和边界。假设分割的阈值为  $(s, t)$ ,则区域 1 目标区域和区域 2 背景区域概率分别为:

$$\begin{cases} p_1 = - \sum_{u=0}^s \sum_{v=0}^t p_{uv} \\ p_2 = - \sum_{u=s+1}^{L-1} \sum_{v=t+1}^{L-1} p_{uv} \end{cases} \quad (14)$$

区域 1 和区域 2 的 2 维熵分别为:

$$\begin{cases} H_1 = - \sum_{u=0}^s \sum_{v=0}^t \frac{p_{uv}}{p_1} \ln \frac{p_{uv}}{p_1} \\ H_2 = - \sum_{u=s+1}^{L-1} \sum_{v=t+1}^{L-1} \frac{p_{uv}}{p_2} \ln \frac{p_{uv}}{p_2} \end{cases} \quad (15)$$

定义分割图像的 2 维熵函数为:

$$H = H_1 + H_2 \quad (16)$$

当 2 维熵函数取最大值时,对应的最优分割阈值  $(s^*, t^*)$  应满足:

$$H(s^*, t^*) = \max \{ H \} \quad (17)$$

式中,  $0 < s < L - 1, 0 < t < L - 1$ 。

为了解决 2 维阈值分割计算时间长及传统智能优化算法常陷入局部最优、收敛速度慢的问题,提出一种基于精英反向的自适应  $t$  分布麻雀算法的 2 维最大熵图像分割方法。

令 ITSSA 的适应度函数为:

$$f_{ITSSA} = H \quad (18)$$

通过融合自适应  $t$  分布的多策略麻雀搜索算法让适应度函数取最大值,此时的  $s$  值和  $t$  值为最优解,即为图像最优分割阈值  $(s^*, t^*)$ 。具体步骤如图 2 所示。

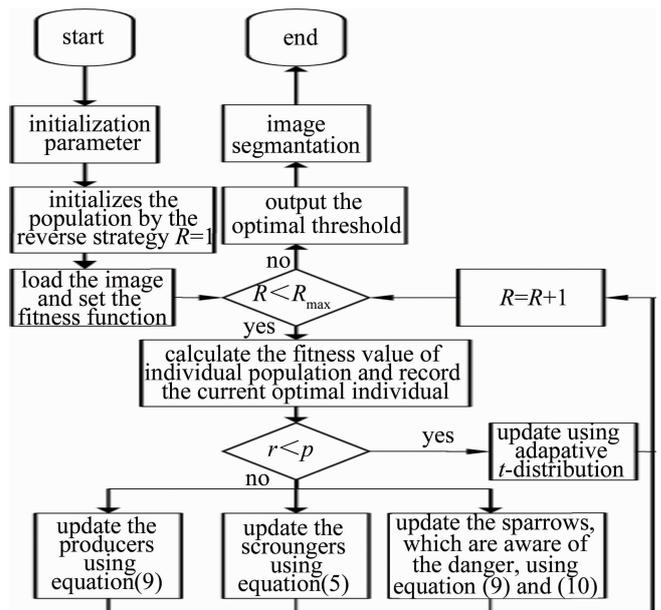


Fig. 2 Flow chart of maximum 2-D entropy based on ITSSA

## 4 实验仿真和结果分析

### 4.1 基准函数测试

为了验证本文中算法的正确性和有效性,使用 SCA,ALO,WOA,GWO,SSA 和 ITSSA 分别测试 15 个基准函数进行比较,充分考察 ITSSA 算法的寻优能力。各算法参量取值见表 1,测试函数名称及参量设置见表 2。表 1 中,  $b$  是对数螺旋形状常数,  $a_s$  是自适应调整常数,  $a_g$  是收敛因子。表 2 中,  $F_1 \sim F_7$  为单峰函数;  $F_8 \sim F_{12}$  为不定维多峰函数;  $F_{13} \sim F_{15}$  为固定维度多峰

Table 1 Algorithm parameter value

algorithm	parameter
WOA	$b = 1$
SCA	$a_s = 2$
GWO	$\max(a_g) = 2, \min(a_g) = 0$
SSA	$D_p = 20\%, R_s = 0.6, D_s = 20\%$
ITSSA	$D_p = 20\%, R_s = 0.6, D_s = 20\%, p = 0.7$

Table 2 Benchmark functions

function	dimension	range	optimal value
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i  x_j  \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max\{ x_i , -1 \leq i \leq n\}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	30	$[-1.28, 1.28]$	0
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	$-418.9829 \times d$
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x) + 10]$	30	$[-5.12, 5.12]$	0
$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x)\right] + 20 + e$	30	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4),$ $y_i = 1 + \frac{x_i + 1}{4},$ $u(x_i, \alpha_5, \alpha_6, m) = \begin{cases} \alpha_6 (x_i - \alpha_5)^m, & (x_i > \alpha_5) \\ 0, & (-\alpha_5 < x_i < \alpha_5) \\ \alpha_6 (-x_i - \alpha_5)^m, & (x_i < -\alpha_5) \end{cases}$	30	$[-50, 50]$	0
$F_{13}(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$	2	$[-50, 50]$	0
$F_{14}(x) = \sum_{i=1}^{11} \left[ \phi_i - \frac{x_1(\varphi_i^2 + \varphi_i x_2)}{\varphi_i^2 + \varphi_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.000307
$F_{15}(x) = 4x_1^2 - 2.14x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316

函数。其中,  $\alpha_5$  和  $\alpha_6$  为  $u$  函数的参量;  $\phi_i$  和  $\varphi_i$  为  $\phi$  数组和  $\varphi$  数组中  $i$  位置的数值;  $e$  为常数。单峰函数仅有一个极大值或极小值, 用于检验算法的收敛速度和精度, 多峰函数具有多个局部或全局最优解, 用于检验算法全局探索能力和开发能力。本文中仿真实验的运行环境为 Core i7 CPU、Windows 10 操作系统, 内存为 8GB, 处理器速度为 3.2GHz。算法在 MATLAB R2020a 软件上运行。在对比测试实验中, 为保证实验结果的客观, 设置各算法种群数量均为 50, 最大迭代次数为 100。

选取各算法在基准函数独立运行 20 次的均值  $M$  和标准差  $D$  作为实验结果, 其中均值反映算法寻优能力和精度, 标准差反映算法的鲁棒性。

本文中的算法 ITSSA 与其它智能算法实验结果对比见表 3, 测试函数收敛情况见图 3。括号中为函数最优值。对于单峰函数  $F_2, F_4, F_5$  和  $F_7$ , ITSSA 寻优均值

和标准差比其它算法高出多个数量级, 说明 ITSSA 具有较高的求解精度、求解速度和鲁棒性。对于  $F_1$  和  $F_3$ , ITSSA 和 SSA 具有相同的标准差, 但是 ITSSA 比 SSA 的均值高出多个数量级。求解  $F_6$  时, SCA 和 ALO 求解能力很差, 与理论最优值存在较大误差。而 ITSSA 在 SSA 的基础上, 相比 WOA 和 GWO 求解表现更好, 且寻优精度变化不大。

对于不定维多峰函数  $F_8$ , 各算法求解能力均不理想, ITSSA 寻优能力仅次于 GWO。对于  $F_9 \sim F_{11}$ , ITSSA 在迭代次数 20 以内可以找到最优解, 不易陷入局部最优, 优于其它算法。对于  $F_{12} \sim F_{14}$ , ITSSA 求解能力最优。对于固定维多峰函数  $F_{15}$ , 各算法都收敛到 -1.3016, ITSSA 标准差最小, 具有最好的稳定性。

综上所述, ITSSA 算法在单峰和多峰函数上的寻优性能均优于 SCA, ALO, WOA, GWO 和基础的 SSA 算法。这主要得益于精英反向策略、自适应  $t$  分布以

Table 3 Comparison of test results of six algorithms

fun	metric	WOA	SCA	GWO	ALO	SSA	ITSSA
$F_1(0)$	$M$	$3.7229 \times 10^{-14}$	$2.5801 \times 10^3$	$8.7660 \times 10^{-4}$	$2.8673 \times 10^3$	$8.3022 \times 10^{-177}$	$1.4325 \times 10^{-264}$
	$D$	$7.2761 \times 10^{-14}$	$1.8959 \times 10^3$	$7.9869 \times 10^{-4}$	$1.6247 \times 10^3$	0.0000	0.0000
$F_2(0)$	$M$	$5.3686 \times 10^{-10}$	4.6605	$6.3202 \times 10^3$	8.4910	$1.6386 \times 10^{-82}$	$2.2242 \times 10^{-122}$
	$D$	$7.8950 \times 10^{-10}$	3.3023	$1.8111 \times 10^3$	0.3625	$8.9751 \times 10^{-82}$	$1.2182 \times 10^{-121}$
$F_3(0)$	$M$	$7.9408 \times 10^4$	$2.6357 \times 10^4$	$2.0940 \times 10^{-2}$	$1.2307 \times 10^4$	$6.6853 \times 10^{-210}$	$5.3945 \times 10^{-241}$
	$D$	$1.7198 \times 10^4$	$9.8335 \times 10^3$	0.7266	$5.2828 \times 10^3$	0.0000	0.0000
$F_4(0)$	$M$	45.1157	68.7204	0.4079	19.3022	$9.1524 \times 10^{-148}$	$3.3763 \times 10^{-220}$
	$D$	26.5841	9.7879	0.2756	4.5386	$8.4629 \times 10^{-148}$	$1.5451 \times 10^{-216}$
$F_5(0)$	$M$	28.6950	$8.5918 \times 10^6$	$3.0882 \times 10^6$	$5.0038 \times 10^5$	$4.2950 \times 10^{-3}$	$2.6764 \times 10^{-4}$
	$D$	0.1326	$7.1854 \times 10^6$	94.1347	$6.0064 \times 10^5$	$7.6155 \times 10^{-4}$	$1.5019 \times 10^{-5}$
$F_6(0)$	$M$	1.3083	$2.3660 \times 10^3$	1.8070	$2.6925 \times 10^3$	$2.0055 \times 10^{-5}$	$1.9577 \times 10^{-5}$
	$D$	0.4415	$2.0276 \times 10^3$	0.6082	$1.4079 \times 10^3$	$5.9475 \times 10^{-5}$	$2.9736 \times 10^{-5}$
$F_7(0)$	$M$	$1.0701 \times 10^{-2}$	2.8404	$9.9002 \times 10^{-3}$	0.8667	$8.6484 \times 10^{-4}$	$7.9608 \times 10^{-5}$
	$D$	$1.0610 \times 10^{-2}$	2.2004	$5.0401 \times 10^{-3}$	0.4107	$7.4407 \times 10^{-4}$	$3.5558 \times 10^{-5}$
$F_8(-12569.487)$	$M$	$-5.7292 \times 10^3$	$-3.5251 \times 10^3$	$-7.8122 \times 10^3$	$-5.7521 \times 10^3$	$-4.3572 \times 10^3$	$-6.1813 \times 10^3$
	$D$	$1.8329 \times 10^3$	$3.4344 \times 10^2$	$1.2021 \times 10^3$	$8.0577 \times 10^2$	$2.1455 \times 10^3$	$1.2825 \times 10^3$
$F_9(0)$	$M$	$2.2737 \times 10^{-10}$	$1.6034 \times 10^2$	26.4377	81.4825	0.0000	0.0000
	$D$	$1.3614 \times 10^{-10}$	67.5549	10.9243	$1.7776 \times 10^2$	0.0000	0.0000
$F_{10}(0)$	$M$	$3.0025 \times 10^{-8}$	20.2426	$7.1491 \times 10^{-3}$	14.6720	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
	$D$	$5.9871 \times 10^{-8}$	5.2773	$1.7111 \times 10^{-3}$	1.3998	0.0000	0.0000
$F_{11}(0)$	$M$	$1.8208 \times 10^{-11}$	22.6283	$2.9440 \times 10^{-3}$	24.1950	0.0000	0.0000
	$D$	$2.4690 \times 10^{-11}$	22.5083	$3.1100 \times 10^{-3}$	25.1536	0.0000	0.0000
$F_{12}(0)$	$M$	$8.1223 \times 10^{-2}$	$1.1808 \times 10^7$	0.1862	$3.4652 \times 10^3$	$4.7649 \times 10^{-6}$	$1.2958 \times 10^{-6}$
	$D$	$5.7852 \times 10^{-2}$	$1.5199 \times 10^7$	0.2013	$1.6216 \times 10^4$	$8.2424 \times 10^{-6}$	$3.4574 \times 10^{-6}$
$F_{13}(0)$	$M$	0.8077	$3.5983 \times 10^7$	1.5182	$1.9085 \times 10^5$	$1.8938 \times 10^{-5}$	$1.3485 \times 10^{-7}$
	$D$	0.2928	$3.8608 \times 10^7$	0.4562	$2.8017 \times 10^5$	$4.3646 \times 10^{-5}$	$2.2025 \times 10^{-7}$
$F_{14}(0.000307)$	$M$	$1.5533 \times 10^{-3}$	$1.1000 \times 10^{-3}$	$3.9820 \times 10^{-3}$	$8.2130 \times 10^{-3}$	$3.2145 \times 10^{-4}$	$3.1795 \times 10^{-4}$
	$D$	$6.0716 \times 10^{-4}$	$3.6909 \times 10^{-4}$	$6.8820 \times 10^{-3}$	$1.3840 \times 10^{-2}$	$1.7059 \times 10^{-5}$	$2.3568 \times 10^{-6}$
$F_{15}(-1.0316)$	$M$	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	$D$	$1.4773 \times 10^{-7}$	$1.5781 \times 10^{-4}$	$1.8250 \times 10^{-7}$	$4.7753 \times 10^{-13}$	$5.9835 \times 10^{-9}$	$1.5673 \times 10^{-15}$

及萤火虫机制,通过这些方法让 ITSSA 算法能有效平衡全局和局部寻优能力,一定程度上避免了算法后期陷入局部最优,让 ITSSA 算法在求解速度、求解精度和稳定性方面均表现出较好的性能,具有较好的寻优能力。

#### 4.2 基于 ITSSA 的 2 维最大熵分割方法

为验证 ITSSA2 维最大熵分割的有效性,选取经典的 rice 图(见图 4)、pepper(见图 5)、伯克利数据集中图 86016(见图 6)和 113016(见图 7)进行对比分析。采用峰值信噪比(peak signal-to-noise ratio, PSNR)、特征相似性(feature similarity index, FSI)<sup>[25]</sup>、各算法运行时间对图像分割质量和算法性能进行评价。其中,峰值信噪比是基于像素均方差的图像相似性的评价指标,PSNR 越大,代表图像失真越少;特征相似性是一

种基于图像低级特征理解图像这一事实的相似性评价指标,FSI 越接近 1,代表图像相似度越强。PSNR 和 FSI 定义如下:

$$\begin{cases} R_{\text{PSNR}} = 20 \lg \left( \frac{255}{e_{\text{RMSE}}} \right) \\ e_{\text{RMSE}} = \sqrt{\frac{\sum_{x=1}^n \sum_{y=1}^n [I(x,y) - I^*(x,y)]^2}{M \times N}} \end{cases} \quad (19)$$

式中, $I(x,y)$ 和 $I^*(x,y)$ 分别代表原图像和分割后的图像。

$$M_{\text{FSIM}} = \frac{\sum_{x \in \Psi} S(x) \cdot C(x)}{\sum_{x \in \Psi} C(x)} \quad (20)$$

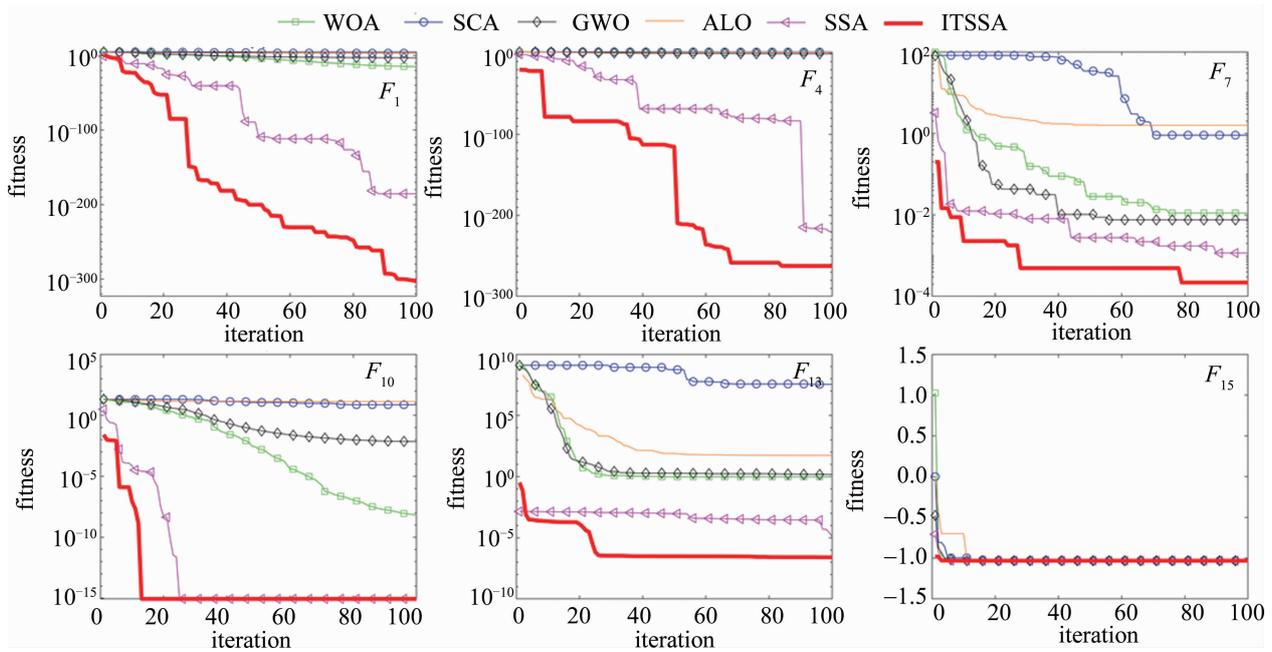


Fig. 3 Convergence graph of the benchmark functions

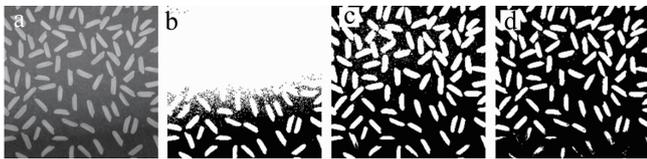


Fig. 4 Segmentation results of image rice

a—original rice. png b—2-D Otsu c—2-D Kapur d—ITSSA 2-D Kapur



Fig. 5 Segmentation results of image pepper

a—original pepper. png b—2-D Otsu c—2-D Kapur d—ITSSA 2-D Kapur

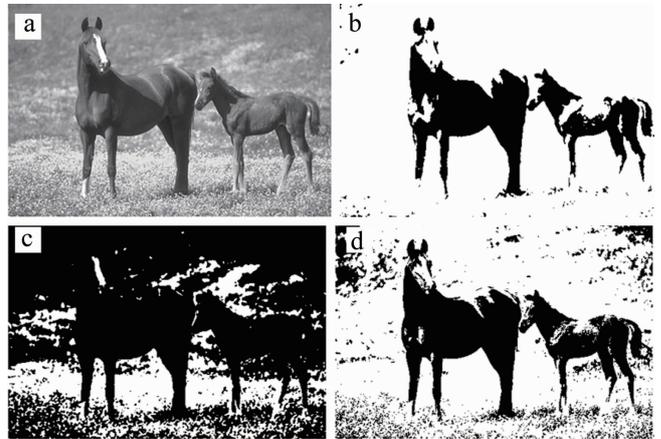


Fig. 7 Segmentation results of image 113016

a—original 113016. png b—2-D Otsu c—2-D Kapur d—ITSSA 2-D Kapur

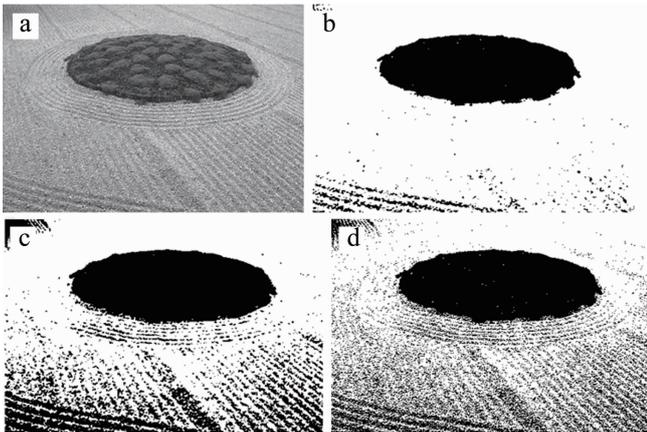


Fig. 6 Segmentation results of image 86016

a—original 86016. png b—2-D Otsu c—2-D Kapur d—ITSSA 2-D Kapur

式中,  $\Psi$  表示整幅图像的空间域,  $C(x)$  代表最大相位,  $S(x)$  代表原图像和分割后图像的相似性值。

$$\begin{cases} C(x) = \max(C_1(x), C_2(x)) \\ S(x) = [F(x)]^\varepsilon \cdot [G(x)]^\delta \\ F(x) = \frac{2C_1(x) \cdot C_2(x) + K_1}{C_1^2(x) + C_2^2(x) + K_1} \\ G(x) = \frac{2G_1(x) \cdot G_2(x) + K_2}{G_1^2(x) + G_2^2(x) + K_2} \end{cases} \quad (21)$$

式中,  $C_1(x)$  和  $C_2(x)$  代表原图像和分割后图像的相位;  $F(x)$  代表图像特征相似性;  $G(x)$  代表图像梯度相似性;  $G_1(x)$  和  $G_2(x)$  代表原图像和分割后图像的梯度幅值;  $\varepsilon, \delta, K_1$  和  $K_2$  均为常数。

在 rice 图的分割实验中, ITSSA 优化的 2 维最大熵在 2 维最大熵基础上提高了峰值信噪比且在结构相似性指标上提升 31.2%。由图 4 可以看出, 本文中算法误判的像素比其它算法少且米粒轮廓最清晰。在 pepper 图的分割实验中, 本文中分割出更清晰的辣椒, 且 3 种指标优于其它算法, 峰值信噪比增加 20%。对于伯克利数据集中的 86016 和 113016, 虽然 2 维 Otsu 比基础的 2 维 Kapur 在峰值信噪比指标和特征相似度表现更好, 但是保留的图片细节大大减少, 耗时过久。本文中提出的算法不仅在 3 种评价指标上表现更好, 改善了 113016 图错分马匹的现象, 也保留了 86016 图和 113016 图更多草地和马匹细节, 分割轮廓上有更好的表现。与此同时, 大大缩短了 2 维最大熵分割耗时时间, 对于  $256 \times 256$  的图像, 将分割图像总时间达到耗时最少的效果(见表 4)。

Table 4 Comparison of values obtained by different segmentation algorithms

image	algorithm	PSNR	FSI	time/s
rice	2-D Otsu	6.8312	0.5264	4.6541
	2-D Kapur	8.7715	0.4829	1.3865
	ITSSA 2-D Kapur	8.9946	0.6336	0.3290
pepper	2-D Otsu	7.8835	0.5664	4.7752
	2-D Kapur	7.4146	0.5709	1.5071
	ITSSA 2-D Kapur	9.3922	0.6063	0.3966
86016	2-D Otsu	9.0520	0.2317	5.8942
	2-D Kapur	8.6222	0.1537	1.7556
	ITSSA 2-D Kapur	9.1398	0.4117	0.3455
113016	2-D Otsu	8.6851	0.4304	7.8510
	2-D Kapur	6.5443	0.4117	2.3696
	ITSSA 2-D Kapur	8.9052	0.4496	0.4070

## 5 结 论

针对麻雀种群角色少、寻优精度不足的现象, 提出了一种基于自适应  $t$  分布的多策略改进麻雀算法。首先利用反向机制引导算法跳出局部最优, 然后将自适应  $t$  分布变异引入位置更新, 发挥其扰动能力。与此同时, 采用精英反向策略提高发现者质量, 扩大搜索区域, 引入萤火虫机制扰动麻雀位置, 提高全局优化能力并且增加种群多样性。融合 ITSSA 和最大 2 维熵分割方法, 缩短了原最大 2 维熵分割方法分割时间, 并将峰值信噪比、特征相似性作为评估标准, 与 2 维 Otsu 法、2 维 Kapur 熵相比, 分割性能更好, 分割出更清晰目标轮廓的同时保留更多细节。实验表明, 基于 ITSSA 的最大 2 维熵图像分割方法在一定程度上解决了麻雀算法后期易陷入局部最优的情况和最大 2 维熵耗时长的

缺点, 具有较好的应用价值, 为群体智能优化算法应用图像分割领域提供参考。在未来研究中, 将尝试寻找并改进出性能更优的群体智能优化算法, 应用图像分割领域提高分割性能效果。

## 参 考 文 献

- [1] SHAPIRO L G, STOCKMAN G C. Computer vision [M]. New Jersey, USA: Prentice Hall, 2001: 279-325.
- [2] WANG X L, YANG C L, XIE G S. Image thresholding segmentation on quantum state space[J]. Entropy, 2018, 20(10): 45-51.
- [3] ZHAO J, ZHU S, YANG W J, et al. An image segmentation algorithm based on density peak clustering[J]. Computer Engineering, 2020, 46(2): 274-278 (in Chinese).
- [4] ZANG F, ZHANG X H. Segmentation algorithm of distance regularization level set based on dislocation theory[J]. Acta Automatica Sinica, 2018, 44(5): 943-952 (in Chinese).
- [5] CAI Q, LIU Y Q, CAO J, et al. A watershed image segmentation algorithm based on self-adaptive marking and interregional affinity propagation clustering[J]. Acta Electronica Sinica, 2017, 45(8): 1911-1918 (in Chinese).
- [6] AGARWAL A, GOEL K. Comparative analysis of digital image for edge detection by using bacterial foraging & canny edge detector [C]//2016 Second International Conference on Computational Intelligence & Communication Technology (CICIT). New York, USA: IEEE, 2016: 125-129.
- [7] ZHU S, GAO R. A novel generalized gradient vector flow snake model using minimal surface and component-normalized method for medical image segmentation [J]. Biomedical Signal Processing and Control, 2016, 26: 1-10.
- [8] MUKHOPADHYAY S, CHANDA B. A multiscale morphological approach to local contrast enhancement[J]. Signal Processing, 2006, 80(4): 685-696.
- [9] TANG K Z, LIU B X, XU H Y, et al. A minimum cross entropy threshold selection method based on genetic algorithm control and decision[J]. Control and Decision, 2013, 28(12): 1805-1810 (in Chinese).
- [10] RAN Q H, GONG Q, WANG K. A new class of image segmentation iterative algorithm based on one-dimensional renyi entropy[J]. Computer and Science, 2014, 41(s2): 91-94 (in Chinese).
- [11] LIU J Z, LI W Q. The automatic thresholding of gray-level pictures via two-dimensional Otsu method [J]. Acta Automatica Sinica, 1993, 19(1): 101-105 (in Chinese).
- [12] ABUTALEB A S. Automatic thresholding of gray-level pictures using two-dimensional entropy[J]. Computer Vision Graphics and Image Processing, 1989, 47(1): 22-32.
- [13] ZHANG X L, YAN H Y, XU Q J. Improved ant colony optimization based medical image thresholding segmentation[J]. Journal of Beijing Jiaotong University, 2016, 40(5): 40-44 (in Chinese).
- [14] ZHANG X, LI Sh L. Research on particle swarm optimization algorithm in image fusion and segmentation[J]. Laser Journal, 2019, 40(11): 84-87 (in Chinese).
- [15] LIU X J, LIU Y L, XU X X. Optimization of multi-threshold Otsu image segmentation by glowworm swarm algorithm with cell membrane mechanism[J]. Journal of Chinese Computer Systems, 2020, 41(2): 410-415 (in Chinese).

- [16] CHEN Ch, XUAN Sh B, LEI H X. Image segmentation based on wolf pack algorithm and 2D maximum entropy[J]. Computer Engineering, 2018, 44(1):233-237(in Chinese).
- [17] YANG X W, YIN H H, HAN X, *et al.* Mesh segmentation based on optimizing extreme learning machine with ant lion optimization[J]. Laser & Optoelectronics Progress, 2020, 57(4):041014(in Chinese).
- [18] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95(5):51-67.
- [19] YONG L Q, LI Y H, JIA W. Literature survey on research and application of sine cosine algorithm[J]. Computer Engineering and Applications, 2020, 56(14):26-34(in Chinese).
- [20] XUE J, SHEN B. A novel swarm intelligence optimization approach: Sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1):22-34.
- [21] HAN F F, LIU S, ZHAO Q H. A self-adaptive flower pollination algorithm based on  $t$ -distribution[J]. Mathematics in Practice and Theory, 2019, 49(2):157-165(in Chinese).
- [22] LIU K, ZHAO L L, WANG H. Whale optimization algorithm based on elite opposition-based and crisscross optimization[J]. Journal of Chinese Computer Systems, 2020, 41(10):2092-2097(in Chinese).
- [23] LI L H, HUA G G. Image segmentation of 2-D maximum entropy based on the improved genetic algorithm[J]. Laser Technology, 2019, 43(1):119-124(in Chinese).
- [24] ZHOU J, WANG L, CHEN X Q. Image segmentation of 2-D maximum entropy based on the improved whale optimization algorithm[J]. Laser Technology, 2021, 45(3):378-385(in Chinese).
- [25] ZHANG L, ZHANG L, MOU X Q, *et al.* FSIM: A feature similarity index for image quality assessment[J]. IEEE Transactions on Image Processing, 2011, 20(8):17-23.