

# 基于改进飞蛾火焰算法 同步优化SVM参数和 特征选择

李小琴<sup>1</sup>, 曲良东<sup>2</sup>

<sup>1</sup>广西民族大学电子信息学院, 广西 南宁

<sup>2</sup>广西民族大学人工智能学院, 广西 南宁

收稿日期: 2023年3月20日; 录用日期: 2023年4月18日; 发布日期: 2023年4月25日

## 摘要

特征选择和参数优化都是提高机器学习分类正确率和效率的重要方法, 本文提出了一种基于改进的飞蛾火焰算法(IMFO)同步优化支持向量机(SVM)参数和特征选择的方法。针对飞蛾火焰算法(MFO)寻优精度较低和容易陷入局部最优的问题, 首先, 利用反馈共享机制增强飞蛾之间的信息交流, 使个体容易脱离局部最优。其次, 引入惯性权重因子改进飞蛾位置更新公式, 增强算法的勘探能力。最后, 将IMFO用于同步优化SVM的参数和特征选择中, 并在12个UCI数据集上进行了特征选择实验, 实验结果表明, 该方法能有效地优化SVM参数和特征子集, 在提高分类准确率的同时, 减少特征数量。

## 关键词

飞蛾火焰算法, 反馈共享机制, 惯性权重因子, 特征选择

# Simultaneous Optimization of SVM Parameters and Feature Selection Based on Improved Moth-Flame Algorithm

Xiaoqin Li<sup>1</sup>, Liangdong Qu<sup>2</sup>

<sup>1</sup>College of Electronic Information, Guangxi Minzu University, Nanning Guangxi

<sup>2</sup>School of Artificial Intelligence, Guangxi Minzu University, Nanning Guangxi

Received: Mar. 20<sup>th</sup>, 2023; accepted: Apr. 18<sup>th</sup>, 2023; published: Apr. 25<sup>th</sup>, 2023

文章引用: 李小琴, 曲良东. 基于改进飞蛾火焰算法同步优化SVM参数和特征选择[J]. 计算机科学与应用, 2023, 13(4): 833-843. DOI: 10.12677/csa.2023.134082

## Abstract

Both feature selection and parameter optimization are important methods to improve the accuracy and efficiency of machine learning classification. In this paper, a method based on Improved Moth-Flame Algorithm (IMFO) is proposed to synchronously optimize support vector machine (SVM) parameters and feature selection. In order to solve the problem of low optimization accuracy and easily fall into local optimization of the Moth-Flame Algorithm (MFO), firstly, a feedback-sharing mechanism is used to enhance the information exchange between moths, so that individuals can easily escape from local optimization. Secondly, the inertia weight factor is introduced to improve the moths' position updating formula to enhance the exploration ability of the algorithm. Finally, IMFO is used for synchronous SVM parameters and feature selection, and feature selection experiments are carried out on 12 UCI datasets. The experimental results show that the proposed method can effectively optimize SVM parameters and feature subsets, improve classification accuracy and reduce the number of features.

## Keywords

Moth-Flame Algorithm, Feedback-Sharing Mechanism, Inertia Weight Factor, Feature Selection

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

如今, 随着科学技术的不断进步, 各个领域都产生了巨大而复杂的信息。为了处理这些海量数据, 数据挖掘和机器学习层出不穷。支持向量机(SVM)是一种常见的机器学习方法, 在处理有限样本、高维和非线性数据等复杂问题时具有独特的优势。支持向量机的性能与核参数和惩罚因子密切相关, 选择合适的参数是提高分类正确率的关键。此外, 原始数据集大多包含冗余和不相关的信息, 这需要很长时间来建模, 并导致性能下降和数据分析困难。因此, 许多研究者使用特征选择等预处理技术从数据集中选择特征子集, 去除噪声、无关和模糊数据, 以提高分类精度。

近年来, 元启发式算法由于其强大高效的性能, 在解决许多组合优化问题方面表现良好。许多研究人员尝试使用元启发式算法来解决特征选择问题, 在尽可能保留信息的同时, 减少原始数据集的维数, 有利于提高模型的准确性和效率。飞蛾火焰优化算法(MFO) [1]算法是 Mirjalili 提出的一种群智能优化算法, 因其具有结构简单、易于操作的优点在许多实际问题中被广泛应用。然而, 对于一些复杂的优化任务, 特别是高维和多模态问题, MFO 可能存在收敛过早、容易陷入局部最优的问题。因此, 许多研究者提出了一些改进策略, 以提升它的性能。本文针对 MFO 算法的不足加以改进, 并应用到特征选择中, 在提升数据集分类正确率的同时, 减少了特征数量。本文的主要研究内容如下:

- 1) 通过引入 Hénon 初始化、反馈共享机制和惯性权重因子到 MFO 算法, 提出了一种改进的飞蛾火焰算法(IMFO), 并在 12 个基准函数上进行测试。

- 2) 将改进的飞蛾火焰算法用于同时优化 SVM 的参数和特征子集, 提高分类的准确率和效率, 并在 12 个 UCI 数据集上进行测试。

## 2. 相关工作

目前, 对 SVM 参数的优化有很多方法。宋玉生等人使用改进的灰狼算法优化 SVM 参数, 并应用到太阳能光伏故障诊断中[2]; Albashish 等人使用改进的生物地理学优化算法优化 SVM 参数[3]; 徐国天等人使用改进的哈里斯鹰算法优化 SVM 参数[4]; 郭辉等人改进的利用鲸鱼优化算法优化 SVM 参数[5]。

此外, MFO 算法在很多领域得到了应用。刘雨豪等人提出了基于飞蛾火焰优化(MFO)算法的模糊控制器设计[6]; 王秀莲等人使用 MFO 算法优化 RBF 神经网络[7]; 徐炜翔等人使用 MFO 算法搜索自主水下航行器路径, 获得能耗最优的无碰路径[8]。他们都取得了很好的效果, 说明飞蛾火焰算法在解决实际工程应用问题上有着较好的能力。

为进一步提高 MFO 算法的优化性能, 不少研究者对该算法进行改进。Li 等人通过引入混沌方法来 MFO 算法的缺点[9]; 何加文等人提出了一种多策略融合改进的飞蛾火焰算法, 并应用于实际工程问题中[10]; Ma 等人引入差分进化提升 MFO 算法跳出局部最优的能力[11]; Elaziz 等人提出一种基于反向学习和差分进化的飞蛾火焰算法用于特征选择[12]。

## 3. 飞蛾火焰优化算法

飞蛾火焰优化算法(MFO)由飞蛾和火焰两种角色构成。飞蛾种群用  $M$  表示, 火焰种群用  $F$  表示, 飞蛾和火焰的适应度值分别用  $OM$  和  $OF$  表示。

飞蛾采用螺旋更新方式, 每只飞蛾对应火焰更新位置方式如公式(1)所示:

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \quad (1)$$

其中,  $M_i$  代表第  $i$  只飞蛾;  $F_j$  代表第  $j$  个火焰;  $D_i$  表示飞蛾与火焰之间的距离,  $D_i = |F_j - M_i|$ ;  $t$  是一个随机数;  $b$  为螺旋形状常数,  $b=1$ 。

每次迭代后火焰按适应度值进行排序, 使用式(2)在迭代过程中自适应地减少火焰的数量:

$$Flame\_no = round\left(n - l * \frac{n-1}{T}\right) \quad (2)$$

其中,  $n$  表示的是当前设定的种群数目,  $l$  表示的算法运行到此的迭代次数,  $T$  代表的算法运行的终止条件, 即最大迭代次数。

飞蛾火焰算法流程为:

- 1) 设置初始参数: 飞蛾和火焰的种群规模、最大迭代次数、初始化种群;
- 2) 计算飞蛾和火焰种群的适应度值;
- 3) 根据公式(1)更新飞蛾的位置;
- 4) 根据公式(2)计算火焰数量, 对火焰进行排序和更新;
- 5) 判断是否达到最大迭代次数, 若达到, 则输出最优结果; 否则返回步骤(2)继续执行操作。

## 4. 支持向量机

支持向量机(SVM)是 Vapnik 为构建统计学习理论而引入的分类模型。它的灵感来自于最大化类之间的几何边界的想法。支持向量机训练阶段的主要目标是使分类线能够将两类区分开, 且类之间的分类间隔最大, 以在特征空间中找到能够很好地分离不同类别的最优分类面。

支持向量机的判别函数为等式(3):

$$f(x) = w^T x + b \quad (3)$$

其中,  $x$  是输入样本,  $b$  是偏差,  $w^T$  表示通过等式(4)计算得到:

$$w = \sum_{i=1}^l \alpha_i y_i x_i, 0 < \alpha_i < c \quad (4)$$

其中,  $\alpha_i$  是拉格朗日乘数,  $c$  是惩罚因子。

SVM 中最常见的核函数是高斯核函数, 如等式(5)所示:

$$k(x_i, x_j) = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)}{2\sigma^2} \quad (5)$$

其中,  $\sigma$  为核参数。

## 5. 改进的飞蛾火焰算法

为了克服原始 MFO 算法寻优精度低和容易陷入局部最优等缺点, 本文提出了一种基于反馈共享和惯性权重因子的飞蛾火焰算法(Improved MFO, IMFO)。首先, 引入 Hénon 初始化以增加种群的多样性, 其次, 利用反馈共享机制增强飞蛾之间的信息交流, 使个体容易脱离局部最优, 最后, 引入惯性权重因子改进飞蛾位置更新公式, 该因子可以显著增强算法的勘探能力。

### 5.1. Hénon 初始化

基本的飞蛾火焰算法采用随机方式初始化飞蛾的位置, 这可能会导致算法在搜索空间中分布不均匀, 影响算法的搜索能力。使用 Hénon 的混沌序列分布可以使种群分布更加均匀, 并增加蝴蝶个体的多样性, 从一定程度上帮助算法跳出局部最优。因此, 本文采用 Hénon 初始化飞蛾种群, 计算方式如下所示[13]:

$$\begin{cases} x(k) = 1 - \gamma(x(k-1))^2 + \varphi y(k-1) \\ y(k) = x(k-1) \end{cases} \quad (6)$$

$$z_i^k = \frac{x_i^k - \varphi_i}{\gamma_i - \varphi_i}, i = 1, 2, \dots, N \quad (7)$$

其中,  $\gamma = 1.4$ ,  $\varphi = 0.3$ ,  $k$  代表混沌迭代次数, 通过式(6)产生[0, 1]区间上  $n$  个不同的混沌序列式(7), 最后通过逆映射得到个体搜索空间变量  $x_i^k$ :

$$x_i^k = lb_i + z_i^k (ub_i - lb_i) \quad (8)$$

其中,  $ub$  和  $lb$  是搜索空间的上、下界。

### 5.2. 反馈共享机制

由于基础的 MFO 算法只能根据火焰的位置进行更新, 缺乏了飞蛾种群之间的信息交流, 容易导致算法陷入局部最优。飞蛾在绕火焰飞行过程中, 彼此之间会不断地进行信息交换, 以找到最亮的火焰。飞蛾向附近随机的一只飞蛾发送信息, 根据随机飞蛾反馈的火焰信息进行比较, 如果比自己发现的火焰亮, 则说明最亮的火焰在随机飞蛾和最优飞蛾周围, 通过正反馈作用, 该飞蛾会移动到两只飞蛾周围; 如果比自己发现的火焰暗, 则说明最亮的火焰不在随机飞蛾周围, 通过负反馈作用, 该飞蛾会移动到最优飞蛾的位置。数学模型描述如下:

$$\alpha = (F_{\max} - F_{\min})r + F_{\min} \quad (9)$$

$$\theta = \frac{1}{1 + \exp(-\lambda)} \quad (10)$$

$$x_i^t = \begin{cases} x_i^t + (g^* - x_i^t)\alpha + (x_r^t - x_i^t)\alpha + \theta, & F(x_r^t) < F(x_i^t) \\ x_i^t + (g^* - x_i^t)\alpha + \theta, & F(x_r^t) \geq F(x_i^t) \end{cases} \quad (11)$$

其中,  $\alpha$  是共享系数,  $F_{\max}$  和  $F_{\min}$  分别是适应度的最大值和最小值,  $r$  是一个 0 到 1 之间的随机数,  $\theta$  是反馈因子,  $\lambda$  表示当前迭代个体数,  $x_r^t$  表示随机飞蛾的位置,  $F$  表示适应度值。

### 5.3. 惯性权重因子

在 MFO 算法中, 飞蛾绕火焰以螺旋方式更新位置, 搜索区域较大, 导致收敛速度较慢, 勘探能力较弱。因此, 本节引入惯性权重因子提升算法的搜索能力, 较大的因子能够使得飞蛾在整个飞行空间内进行搜索, 较小的因子能够使得飞蛾在局部区域内精准搜索。该因子通过如下公式更新:

$$w(t) = w_{end} \cdot (w_{start} - w_{end}) \cdot (T_{max} - t) / T_{max} \quad (12)$$

其中,  $w_{start} = 0.8$ ,  $w_{end} = 0.4$ 。

引入权重因子之后, 飞蛾的位置更新公式为:

$$D_i = |F_j - w \cdot M_i| \quad (13)$$

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + w \cdot F_j \quad (14)$$

### 5.4. 改进的飞蛾火焰算法流程

改进的飞蛾火焰算法流程如下:

- 1) 设置初始参数: 飞蛾和火焰的种群规模、最大迭代次数;
- 2) 使用公式(8)初始化种群;
- 3) 计算飞蛾和火焰种群的适应度值;
- 4) 随机选择一个飞蛾  $x_r$ , 比较当前飞蛾  $x_i$  与随机飞蛾的适应度值;
- 5) If  $\text{fitness}(x_r) < \text{fitness}(x_i)$ , 使用公式(11)中的正反馈更新飞蛾位置, 否则, 使用负反馈更新飞蛾位置;
- 6) 使用引入惯性权重因子的公式(14)更新飞蛾位置;
- 7) 使用公式(2)计算火焰的数量, 对火焰进行排序和更新;
- 8) 判断是否达到最大迭代次数, 若达到, 则输出最优结果; 否则返回步骤(3)继续执行操作。

## 6. IMFO-SVM 同步优化参数和特征子集

特征选择是一个二进制优化问题, 必须将 CESDAVO 的连续值转换为相应的二进制值。也就是说, 每个特征子集都可以被视为飞蛾的位置, 飞蛾向量的维数是特征子集的长度。矢量的每个单元可以包含两个值 1 或 0, 其中值 1 表示选择了相应的特征, 值 0 表示未选择该特征。

该转换是通过使用 Sigmoid 函数来实现的, 转换公式如等式(15)所示:

$$x_{i,j} = \begin{cases} 1, & \text{if } \text{Sig}(y_{i,j}) > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

特征选择的目标是在保证较高分类正确率的前提下选择最少的特征子集。因此, 本文使用适应度函数(16)来评估每个解:

$$\text{Fitness} = r_1 E + r_2 \frac{|S|}{|N|} \quad (16)$$

其中,  $E$  表示 SVM 分类器给出的错误率,  $S$  是所选特征的数量,  $N$  是原始数据集的特征数量,  $r_1 \in [0,1]$ ,  $r_2 = 1 - r_1$  是惩罚因子。本文实验中  $r_1 = 0.99$ ,  $r_2 = 0.01$ 。

使用 IMFO 算法优化 SVM 特征选择的具体步骤如下:

- 1) 输入数据, 划分数据集, 并将数据集进行归一化处理;
- 2) 创建 SVM 模型, 随机生成 SVM 参数, 初始化种群, 并进行编码: 飞蛾种群维度的前两维采用实数编码用于优化 SVM 的参数  $c$  和  $\sigma$ , 其他维度进行二进制编码用于选择特征子集;
- 3) 利用式(16)的适应度函数计算适应度值;
- 4) 使用 IMFO 算法优化搜索 SVM 的最优参数( $c, \sigma$ )和特征子集;
- 5) 将优化后的参数和特征子集的训练集一起输入到 SVM 模型中进行训练;
- 6) 计算适应度值, 适应度值最小的子集即为具有最大分类正确率和最少特征数量的最优子集;
- 7) 用所选特征子集的训练集对训练的 SVM 进行测试;
- 8) 判断是否满足算法终止条件, 如果是, 则进行结果评估并输出最终结果, 否则返回步骤(3)继续运行。

基于 IMFO 算法优化 SVM 特征选择的流程图如下图 1 所示。

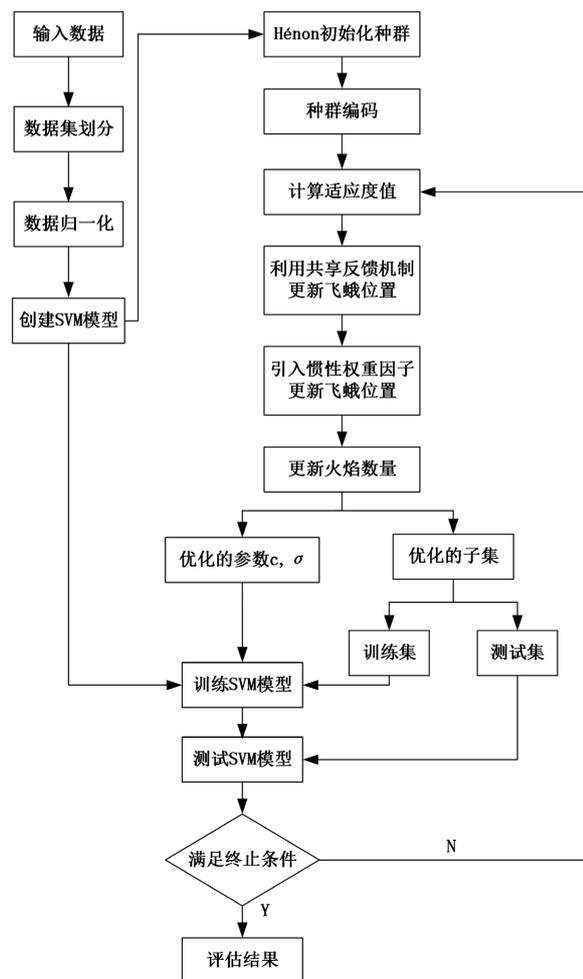


Figure 1. Flow chart of IMFO-SVM

图 1. IMFO-SVM 流程图

## 7. 仿真实验

### 7.1. 改进飞蛾火焰算法测试实验

#### 7.1.1. 测试函数

为测试 IMFO 算法的能力, 本章采用 12 个函数对该算法进行测试, 其中 F1~F4 是单峰型, 它们只有一个全局最优解, 用于验证算法的收敛速度; F5~F8 是多峰型, 它们具有局部最优解, 可用于判断算法避免陷入局部最优的能力。F9~F12 是固定维多峰型, 固定维多峰函数提供了比其他基准函数更多的搜索空间, 用于判断算法的全局搜索能力。测试函数如表 1 所示。

**Table 1.** Test functions

**表 1.** 测试函数

函数	名称	函数	名称
F1	Sphere	F7	Griewank
F2	Schwefel's function 2.21	F8	Generalized penalized 2
F3	Schwefel's function 1.2	F9	Kowalik
F4	Schwefel's function 2.22	F10	Goldstein-Price
F5	Rastrigin	F11	Hartman 2
F6	Ackley	F12	Generalized penalized 1

#### 7.1.2. 实验结果与分析

为测试引入的改进策略是否有效, 分别在 12 个函数的 30 维度和 100 维度上对 IMFO 与 MFO 进行测试, 算法的迭代次数均设为 500, 在每个函数上运行 30 次, 并记录每个函数上适应度值的最优值、平均值和标准差, 实验结果如表 2 所示。

**Table 2.** The result of function test

**表 2.** 函数测试结果

函数	算法	Dim = 30			Dim = 100		
		最优值	平均值	标准差	最优值	平均值	标准差
F1	MFO	1.259E+00	4.341E+03	6.786E+03	1.376E-01	1.264E+00	1.063E+00
	IMFO	<b>3.11E-159</b>	<b>3.97E-150</b>	<b>2.13E-149</b>	<b>2.44E-160</b>	<b>3.54E-154</b>	<b>1.08E-153</b>
F2	MFO	1.966E-01	3.313E+01	1.853E+01	8.656E-02	3.440E+01	1.991E+01
	IMFO	<b>6.644E-85</b>	<b>2.432E-80</b>	<b>6.405E-80</b>	<b>3.111E-83</b>	<b>5.282E-81</b>	<b>1.005E-80</b>
F3	MFO	6.356E+03	1.926E+04	1.056E+04	2.094E+03	1.447E+04	9.538E+03
	IMFO	<b>3.79E-150</b>	<b>4.96E-141</b>	<b>1.88E-140</b>	<b>2.68E-149</b>	<b>1.05E-143</b>	<b>3.86E-143</b>
F4	MFO	5.129E+01	6.675E+01	7.727E+00	2.763E+01	4.125E+01	9.304E+00
	IMFO	<b>1.394E-77</b>	<b>5.313E-73</b>	<b>1.603E-72</b>	<b>4.190E-78</b>	<b>1.527E-75</b>	<b>2.654E-75</b>
F5	MFO	9.954E+01	1.609E+02	3.735E+01	5.278E+01	1.363E+02	3.341E+01
	IMFO	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>

Continued

F6	MFO	5.446E-01	1.387E+01	7.488E+00	1.905E-01	1.050E+01	9.430E+00
	IMFO	<b>8.882E-16</b>	<b>8.882E-16</b>	<b>0.000E+00</b>	<b>8.882E-16</b>	<b>8.882E-16</b>	<b>0.000E+00</b>
F7	MFO	4.414E-01	1.293E+01	3.907E+01	2.190E-01	9.832E+00	2.738E+01
	IMFO	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>
F8	MFO	4.584E+00	1.367E+07	7.487E+07	<b>4.269E-01</b>	3.505E+00	2.270E+00
	IMFO	<b>2.523E+00</b>	<b>2.890E+00</b>	<b>9.387E-02</b>	2.459E+00	<b>2.852E+00</b>	<b>1.277E-01</b>
F9	MFO	4.340E-04	1.947E-03	3.742E-03	6.338E-04	8.732E-04	2.256E-04
	IMFO	<b>3.413E-04</b>	<b>6.560E-04</b>	<b>1.416E-04</b>	<b>3.140E-04</b>	<b>5.280E-04</b>	<b>1.252E-04</b>
F10	MFO	3.000E+00	3.004E+00	5.027E-03	3.000E+00	3.001E+00	4.787E-04
	IMFO	<b>3.000E+00</b>	<b>3.000E+00</b>	<b>1.363E-15</b>	<b>3.000E+00</b>	<b>3.000E+00</b>	<b>1.591E-15</b>
F11	MFO	-2.45E+00	-1.88E+00	2.584E-01	-2.96E+00	-2.26E+00	3.299E-01
	IMFO	<b>-3.32E+00</b>	<b>-3.21E+00</b>	<b>1.065E-01</b>	<b>-3.32E+00</b>	<b>-3.22E+00</b>	<b>4.518E-02</b>
F12	MFO	-3.84E+00	-2.38E+00	<b>7.791E-01</b>	-4.96E+00	-3.06E+00	7.73E-01
	IMFO	<b>-1.02E+01</b>	<b>-7.98E+00</b>	3.204E+00	<b>-1.02E+01</b>	<b>-8.05E+00</b>	<b>2.88E+00</b>

在单峰函数测试中, IMFO 在 4 个测试函数上的 30、100 维度上均获得了最小的适应度、平均值和标准差, 这表明 IMFO 算法具有更强的寻优能力和稳定性。在多峰函数测试中, IMFO 在 F5、F6、F7 上同样获得了最小的适应度值、平均值和标准差, 在 F8 的 30 维度上得到了了最优的适应度值、平均值和标准差, 在 100 维上虽然没有得到最小的适应度值, 但得到了最小的平均值和标准差, 说明 IMFO 在多峰函数上也比 MFO 有更好的表现。在固定维多峰函数上, IMFO 算法也在 30 和 100 维度上均获得了最小的最优值、平均值和标准差, 说明 IMFO 算法具有更好的全局搜索能力。总的来说, 这些实验结果表明, 引进的改进策略大大提升了原始算法的寻优能力, 并且随着维度的增加, IMFO 算法的寻优能力越来越好, 说明改进的算法在解决高维和复杂问题上也具有强大的能力。

## 7.2. 特征选择实验

### 7.2.1. 数据集

UCI 是一个广泛用于数据分析和分类的机器学习数据库([UCI Machine Learning Repository: Data Sets](https://archive.ics.uci.edu/))。本文使用 12 个数据集来评估 IMFO-SVM 的性能。

### 7.2.2. 评价指标

以下标准用于评估不同算法的性能:

- 1) 适应度值: 它们是从适应度函数中获得的, 适应度函数表示算法收敛的速度和值;
- 2) 分类正确率: 它是分类器给出的指标, 描述了算法选择特征集的准确程度。该值越大, 分类效率越好;
- 3) 特征约简率: 这是一个描述特征减少率的指标, 由  $R = \frac{N-S}{N}$  计算得到。其中,  $N$  是原始数据集的特征数量,  $S$  是所选特征数量。该值越大, 说明特征约简效率越高。

### 7.2.3. 实验结果与分析

为测试 IMFO 在同步优化中的有效性, 使用了四个优化算法进行比较, 包括差分进化算法(DE)、灰

狼优化算法(GWO)、粒子群算法(PSO)和鲸鱼优化算法(WOA)。为保证实验的公平性,种群的大小和最大迭代次数分别设置为 20 和 100。所有算法在每个数据集上运行 20 次,分别计算适应度值、分类正确率和特征选择率的最优值和平均值。

表 3 显示了 20 次运行中每个算法获得的适应度值的最优值和平均值,从中我们可以看出,基于 MFO 的特征选择方法在 12 个数据集中均获得了最优的适应值,即适应度函数的最小值。具体而言,IMFO 算法在 4 个数据集上适应度值优于 DE,在 8 个数据集上与之并列第一;在 8 个数据集上适应度值优于 GWO,在 4 个数据集上与之并列第一;在 5 个数据集上适应度值优于 PSO,在 7 个数据集上与之并列第一;在 10 个数据集上适应度值优于 WOA,在 2 个数据集上与之并列第一。这说明该方法在同步优化 SVM 参数和特征子集问题上的寻优能力明显优于其他算法。此外,IMFO 算法在除了 Vote 和 Wine 之外的 10 个数据集中都获得了最优的平均适应度值,这意味着它在该问题的寻优过程中比 GWO、PSO、WOA 和 MFO 具有更好的稳定性。

**Table 3.** Comparison of fitness values of different optimization algorithms

**表 3.** 不同优化算法的适应度值比较

数据集	最优值					平均值				
	DE	GWO	PSO	WOA	IMFO	DE	GWO	PSO	WOA	IMFO
Australian	0.144	0.223	0.144	0.172	<b>0.144</b>	0.168	0.230	0.178	0.219	<b>0.159</b>
Breast cancer	0.000	0.036	0.036	0.050	<b>0.000</b>	0.043	0.047	0.050	0.054	<b>0.033</b>
BreatEW	0.009	0.000	0.009	0.018	<b>0.000</b>	0.016	0.009	0.019	0.018	<b>0.011</b>
HeartEW	0.074	0.129	0.092	0.202	<b>0.055</b>	0.020	0.138	0.147	0.220	<b>0.119</b>
IonosphereEW	0.014	0.042	0.014	0.028	<b>0.014</b>	0.025	0.042	0.031	0.035	<b>0.018</b>
Lymphography	0.103	0.137	0.103	0.137	<b>0.103</b>	0.147	0.154	0.140	0.137	<b>0.135</b>
Segmentation	0.165	0.330	0.189	0.307	<b>0.165</b>	0.281	0.413	0.304	0.318	<b>0.258</b>
SonarEW	0.048	0.072	0.048	0.072	<b>0.048</b>	0.070	0.085	0.070	0.085	<b>0.058</b>
Vehicle	0.281	0.709	0.270	0.621	<b>0.258</b>	0.463	0.724	0.474	0.680	<b>0.417</b>
Vote	0.011	0.000	0.000	0.011	<b>0.000</b>	0.014	<b>0.006</b>	0.010	0.011	0.011
Wine	0.000	0.000	0.000	0.000	<b>0.000</b>	0.023	<b>0.000</b>	0.040	0.014	0.013
Zoo	0.000	0.000	0.000	0.000	<b>0.000</b>	0.000	0.000	0.000	0.000	<b>0.000</b>

各算法在 12 个数据上分类正确率的最优值和平均值如表 4 所示。实验结果表明,IMFO 算法在 12 个数据集中分类正确率的最优值均是最高。具体而言,IMFO 在 5 个数据集上的最优分类正确率高于 DE,在 7 个数据集上与 DE 并列第一;在 8 个数据集上的正确率高于 GWO,在 4 个数据集上与 GWO 并列第一;在 5 个数据集上的正确率高于 PSO,在 7 个数据集上与 PSO 并列第一;在 9 个数据集上的正确率高于 WOA,在 3 个数据集上与 WOA 并列。这表明 IMFO 分类正确率的最优值明显优于其他是个算法。此外,IMFO 在除 Breast cancer 和 Vote 之外的 10 个数据集上的平均分类正确率均高于 DE、GWO、PSO 和 WOA。这说明该算法不仅有效地提高了数据集的分类正确率,并且具有很好的稳定性。

算法在 12 个数据集上的特征选择率如表 5 所示。IMFO 算法在 8 个数据集上特征约简率的最优值高于其他算法,分别达到了 78.6%、77.8%、76.9%、26.5%、66.7%、45%、77.8%、53.8%,这说明 IMFO 算法能大大地约简原始数据集的特征,且相较于其他算法有更好的特征约简能力。综合 IMFO 算法获得

的分类正确率, 证明了该算法可以在约简特征和提升分类正确率上达到很好的平衡。

**Table 4.** Comparison of classification accuracy rate of different optimization algorithms  
**表 4.** 不同优化算法的分类正确率比较

数据集	最优值					平均值				
	DE	GWO	PSO	WOA	IMFO	DE	GWO	PSO	WOA	IMFO
Australian	0.855	0.775	0.855	0.820	<b>0.855</b>	0.831	0.768	0.821	0.818	<b>0.839</b>
Breast cancer	0.971	0.964	0.964	0.970	<b>0.978</b>	0.857	<b>0.953</b>	0.950	0.920	0.934
BreatEW	0.991	1.000	0.991	0.991	<b>1.000</b>	0.984	0.981	0.981	0.981	<b>0.989</b>
HeartEW	0.926	0.870	0.907	0.870	<b>0.944</b>	0.881	0.861	0.852	0.851	<b>0.878</b>
IonosphereEW	0.986	0.957	0.986	0.967	<b>0.986</b>	0.975	0.957	0.969	0.967	<b>0.981</b>
Lymphography	0.897	0.862	0.897	0.872	<b>0.897</b>	0.852	0.845	0.859	0.853	<b>0.862</b>
Segmentation	0.833	0.667	0.810	0.767	<b>0.833</b>	0.717	0.583	0.693	0.689	<b>0.738</b>
SonarEW	0.951	0.927	0.951	0.942	<b>0.951</b>	0.929	0.915	0.929	0.923	<b>0.941</b>
Vehicle	0.716	0.254	0.728	0.373	<b>0.734</b>	0.533	0.254	0.521	0.314	<b>0.579</b>
Vote	0.989	1.000	1.000	1.000	<b>1.000</b>	0.986	<b>0.994</b>	0.990	0.979	0.989
Wine	1.000	1.000	1.000	1.000	<b>1.000</b>	0.977	1.000	0.960	1.000	<b>1.000</b>
Zoo	1.000	1.000	1.000	1.000	<b>1.000</b>	1.000	1.000	1.000	1.000	<b>1.000</b>

**Table 5.** Comparison of feature reduction rate of different optimization algorithms  
**表 5.** 不同优化算法的特征约简率比较

数据集	最优值					平均值				
	DE	GWO	PSO	WOA	IMFO	DE	GWO	PSO	WOA	IMFO
Australian	0.786	0.357	0.786	0.357	<b>0.786</b>	0.525	0.321	0.464	0.357	<b>0.564</b>
Breast cancer	0.667	0.444	0.556	0.333	<b>0.778</b>	0.411	0.278	0.544	0.333	<b>0.544</b>
BreatEW	0.267	0.133	0.400	0.033	0.167	0.147	0.083	0.297	0.033	0.085
HeartEW	0.615	0.462	0.692	0.385	<b>0.769</b>	0.538	0.385	0.569	0.346	<b>0.485</b>
IonosphereEW	0.000	0.000	0.178	0.118	<b>0.265</b>	0.038	0.226	0.000	0.059	0.149
Lymphography	0.611	0.222	0.556	0.167	0.333	0.272	0.167	0.378	0.139	0.183
Segmentation	0.667	0.222	0.556	0.333	<b>0.667</b>	0.378	0.111	0.333	0.222	<b>0.400</b>
SonarEW	0.233	0.333	0.367	0.317	<b>0.450</b>	0.145	0.192	0.233	0.317	0.153
Vehicle	0.722	0.056	0.778	0.000	<b>0.778</b>	0.450	0.056	0.486	0.000	<b>0.522</b>
Vote	0.500	0.063	0.375	0.125	0.188	0.178	0.063	0.225	0.063	0.053
Wine	0.462	0.308	0.538	0.462	<b>0.538</b>	0.331	0.308	0.400	0.304	<b>0.462</b>
Zoo	0.188	0.063	0.375	0.000	0.125	0.047	0.031	0.119	0.000	0.013

## 8. 总结

本文提出了一种改进的飞蛾火焰算法(IMFO)。首先, 使用 Hénon 混沌机制对种群初始化, 有效增加

了飞蛾种群的多样性;其次,飞蛾在迭代过程中随机挑选一只飞蛾建立反馈机制,增强飞蛾之间信息交流,通过正负反馈原理提升了算法逃离局部最优的能力;最后,引入惯性权重因子提升算法后期的勘探能力,提升算法寻优能力并加快算法收敛速度。通过在基准函数上的实验结果表明,IMFO算法有效提升了MFO算法的寻优能力,能够得到比原始算法精度更高的最优解。

在此基础上,将IMFO算法转换为二进制编码,并用于同时优化特征子集和SVM的参数。使用12个UCI数据集对算法进行测试,并与其他4个算法(GWO、PSO、WOA和MFO)进行比较。实验结果表明,IMFO算法能在保证高分类准确率的前提下,降低特征数量,优于其他算法的特征选择能力,同时具有更好的稳定性。

## 基金项目

广西自然科学基金(GUIKE AD18126010),广西高校中青年教师科研基础能力提升项目(2022KY0164)。

## 参考文献

- [1] Wang, G.G. (2018) Moth Search Algorithm: A Bio-Inspired Metaheuristic Algorithm for Global Optimization Problems. *Memetic Computing*, **10**, 151-164. <https://doi.org/10.1007/s12293-016-0212-3>
- [2] 宋玉生, 刘光宇, 朱凌, 等. 改进的灰狼优化算法在 SVM 参数优化中的应用[J]. 传感器与微系统, 2022, 41(9): 151-155. [https://doi.org/10.13873/J.1000-9787\(2022\)09-0151-05](https://doi.org/10.13873/J.1000-9787(2022)09-0151-05)
- [3] Albashish, D., Hammouri, A.I., Braik, M., Atwan, J. and Sahran, S. (2021) Binary Biogeography-Based Optimization Based SVM-RFE for Feature Selection. *Applied Soft Computing Journal*, **101**, Article ID: 107026. <https://doi.org/10.1016/j.asoc.2020.107026>
- [4] 徐国天, 刘猛猛. 基于改进哈里斯鹰算法同步优化特征选择的恶意软件检测方法[J]. 信息安全, 2021, 21(12): 9-18. [https://kns.cnki.net/kcms2/article/abstract?v=3uoqIhG8C44YLTIOAiTRKibYIV5Vjs7iJTKGjg9uTdeTsOI\\_ra5\\_XQ19-tXiS13CsEu2v6XQyM2asg6idWUB-hYAtUT2IYjI&uniplatform=NZKPT](https://kns.cnki.net/kcms2/article/abstract?v=3uoqIhG8C44YLTIOAiTRKibYIV5Vjs7iJTKGjg9uTdeTsOI_ra5_XQ19-tXiS13CsEu2v6XQyM2asg6idWUB-hYAtUT2IYjI&uniplatform=NZKPT)
- [5] 郭辉, 付接递, 李振东, 严岩, 李斌. 基于改进鲸鱼算法优化 SVM 参数和特征选择[EB/OL]. 吉林大学学报(工学版): 1-22. <https://doi.org/10.13229/j.cnki.jdxbgxb20211348>, 2022-04-07.
- [6] 刘雨豪, 廖平. 基于 MFO 算法的无刷直流电机模糊控制设计[J]. 仪表技术与传感器, 2021, 459(4): 107-111.
- [7] 王秀莲, 杨帅, 高宏伟, 等. 基于 MFO 优化 RBF 神经网络的舵机控制与仿真[J]. 信息技术与信息化, 2023, 275(2): 179-183.
- [8] 徐炜翔, 朱志宇. 基于飞蛾火焰算法的 AUV 三维全局路径规划[J]. 上海理工大学学报, 2021, 43(2): 148-155. <https://doi.org/10.13255/j.cnki.jusst.20200407005>
- [9] Li, H.G., Liu, J.Y., Chen, L., Bai, J.B., Sun, Y.Y. and Lu, K. (2019) Chaos-Enhanced Moth-Flame Optimization Algorithm for Global Optimization. *Journal of Systems Engineering and Electronics*, **30**, 1144-1159. <https://doi.org/10.21629/JSEE.2019.06.10>
- [10] 何加文, 许贤泽, 高波. 多策略融合改进的飞蛾火焰优化算法[EB/OL]. 北京航空航天大学学报: 1-12. <https://doi.org/10.13700/j.bh.1001-5965.2022.0707>, 2022-11-08.
- [11] Ma, L., Wang, C., Xie, N.G., Shi, M., Ye, Y. and Wang, L. (2021) Moth-Flame Optimization Algorithm Based on Diversity and Mutation Strategy. *Applied Intelligence*, **51**, 5836-5872. <https://doi.org/10.1007/s10489-020-02081-9>
- [12] Elaziz, M.A., Ewees, A.A., Ibrahim, R.A. and Lu, S.F. (2020) Opposition-Based Moth-Flame Optimization Improved by Differential Evolution for Feature Selection. *Mathematics and Computers in Simulation*, **168**, 48-75. <https://doi.org/10.1016/j.matcom.2019.06.017>
- [13] 李守玉, 何庆, 杜逆索. 混沌反馈共享和群体协同效应的蝴蝶优化算法[J]. 计算机科学与探索, 2022, 16(7): 1661-1672.