Advances in Applied Mathematics 应用数学进展, 2026, 15(6), 26-38
Published Online June 2026 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2026.156262

Hans汉斯

改进水母优化算法及其在多阈值图像分割中的应用

郝　岩[1,2*]，侯宇超[3]

[1]太原师范学院数学与统计学院，山西　晋中
[2]智能优化计算与区块链技术山西省重点实验室，山西　晋中
[3]山西师范大学密码学与数据安全山西省重点实验室，山西　太原

收稿日期：2026年4月29日；录用日期：2026年5月23日；发布日期：2026年6月2日

摘　要

针对水母优化算法(Jellyfish Search Optimizer, JSO)在高维优化问题中收敛速度慢、易陷入局部最优的缺陷，提出一种改进水母优化算法(Improved Jellyfish Search, IJS)。在JSO基础上引入自适应时间控制策略以平衡全局探索与局部开发能力，设计精英引导机制提升收敛精度与寻优效率，并采用硬边界约束保证迭代解在可行域内的有效性。为验证IJS算法的性能，本文首先基于6个标准测试函数，将IJS与JSO、灰狼优化算法、粒子群优化算法、鲸鱼优化算法进行寻优性能对比；其次将IJS应用于图像多阈值分割，基于Otsu类间方差最大化准则构建适应度函数，将FSIM、SSIM、PSNR作为评价指标，与JSO及传统Otsu算法对比分割效果。实验结果表明，IJS算法具有更优的寻优精度、收敛速度和稳定性，可有效应用于函数寻优与图像多阈值分割等复杂优化场景。

关键词

水母优化算法，自适应时间控制，精英引导，多阈值图像分割

Improved Jellyfish Search Optimizer Algorithm and Its Application in Multi-Threshold Image Segmentation

Yan Hao[1,2*], Yuchao Hou[3]

[1]School of Mathematics and Statistics, Taiyuan Normal University, Jinzhong Shanxi
[2]Shanxi Key Laboratory of Intelligent Optimization Computing and Blockchain Technology, Jinzhong Shanxi
[3]Shanxi Key Laboratory of Cryptography and Data Security, Shanxi Normal University, Taiyuan Shanxi

[*]通讯作者。

文章引用: 郝岩, 侯宇超. 改进水母优化算法及其在多阈值图像分割中的应用[J]. 应用数学进展, 2026, 15(6): 26-38.
DOI: 10.12677/aam.2026.156262

Received: April 29, 2026; accepted: May 23, 2026; published: June 2, 2026

Abstract

To address the shortcomings of the Jellyfish Search Optimizer (JSO), including slow convergence and a tendency to become trapped in local optima when solving high-dimensional optimization problems, an Improved Jellyfish Search (IJS) algorithm is proposed. Based on the original JSO, an adaptive time control strategy is introduced to balance global exploration and local exploitation, an elite-guidance mechanism is designed to improve convergence accuracy and optimization efficiency, and a hard boundary constraint is adopted to ensure the validity of iterative solutions within the feasible domain. To verify the performance of the proposed IJS algorithm, six benchmark functions are first employed to compare its optimization performance with that of JSO, Grey Wolf Optimizer, Particle Swarm Optimization, and Whale Optimization Algorithm. Furthermore, IJS is applied to multi-threshold image segmentation, where the fitness function is constructed based on Otsu's maximum between-class variance criterion. FSIM, SSIM, and PSNR are selected as evaluation metrics, and the segmentation performance is compared with that of JSO and the conventional Otsu method. Experimental results show that the IJS algorithm achieves superior optimization accuracy, convergence speed, and stability, and can be effectively applied to complex optimization scenarios such as function optimization and multilevel image thresholding.

Keywords

Jellyfish Search Optimizer, Adaptive Time Control, Elite Guidance, Multilevel Image Thresholding

1. 引言

群智能优化算法源于对自然界生物群体行为的模拟，因其具有无需梯度信息、鲁棒性强、实现简单、适配性广等优势，已被广泛应用于函数寻优、参数优化、路径规划、图像分割等多个领域[1]。近年来，各类新型群智能算法不断涌现，其中 2021 年 Chou 等人提出的水母优化算法，通过模拟水母在海洋中的洋流运动、被动运动和主动运动三种行为，实现优化问题的求解，具有参数设置少、易实现的特点[2]。然而，原始 JSO 算法仍存在明显不足，首先时间控制系数采用线性衰减模式，难以平衡算法前期的全局探索能力与后期的局部开发能力；其次缺乏对精英个体信息的利用，收敛精度和收敛速度有待提升，在高维复杂优化问题中易陷入局部最优[3]。针对这些不足，当前也有很多 JSO 的改进算法，如基于随机榜样学习的洋流运动策略[4]，引入记忆函数和混合策略改进水母的位置更新机制[5]，引入指数递减惯性权重，扩大算法全局搜索范围等[6]。

图像多阈值分割是计算机视觉、图像处理领域的核心任务之一，其核心是通过选取最优阈值将图像灰度级划分为多个区域，实现目标与背景的有效分离，广泛应用于医学影像分析、遥感图像处理、目标检测等场景[7]。Otsu 算法作为经典的阈值选取方法，通过最大化类间方差实现最优阈值求解，但传统 Otsu 算法采用遍历方式搜索阈值，当阈值维度提升时，时间复杂度呈指数增长，难以适用于高维阈值分割场景[8] [9]。群智能算法为高维阈值寻优提供了高效解决方案，已成为图像多阈值分割的研究热点。

针对原始 JSO 算法的缺陷及图像分割的实际需求，本文提出一种改进水母优化算法，通过三大改进策略提升算法性能，并从函数寻优和图像多阈值分割两个维度开展对比实验，验证算法的有效性和实用性。本文的主要研究工作如下：

1) 对原始 JSO 算法进行改进，引入自适应时间控制、精英引导策略和硬边界约束，解决原始算法中过早收敛、收敛速度慢、精度低的问题；

2) 基于 6 个标准测试函数，设计函数寻优实验，对比 IJS 与 JSO 及其他经典群智能算法的寻优性能，从最优值、均值、标准差、和时间开销四个维度进行全面分析；

3) 将 IJS 算法应用于图像多阈值分割，基于 Otsu 类间方差最大化准则构建适应度函数，设置 2、4、6、8、10 维阈值场景，对比 IJS 与 JSO、传统 Otsu 算法的分割效果，通过多维度评价指标验证算法的分割性能。

2. 相关理论基础

2.1. 水母优化算法

JSO 算法的基本思想是模拟水母在海洋中的觅食行为，把优化过程抽象为三条规则。其一，水母要么跟随洋流，要么在群体内部运动，并由时间控制机制决定两者切换；其二，水母会向食物更丰富的位置移动；其三，某位置的“食物量”由该位置对应的目标函数值决定。其核心原理和数学模型如下：

1) 时间控制系数：用于控制水母三种运动行为的概率，采用线性衰减模式，公式为：

$$c(t)=\left|\left(1-\frac{t}{\mathrm{Max}_{\mathrm{iter}}}\right)\times\left(2\times rand(0,1)-1\right)\right| \tag{1}$$

其中，$c_0=0.5$ 为初始时间控制系数，t 为当前迭代次数，T 为最大迭代次数。当 $c(t)\geq 0.5$ 时，个体倾向于跟随洋流执行全局探索；当 $c(t)<0.5$ 时，个体转入群内运动；而在群内运动中，又通过 $1-c(t)$ 来调节主动运动与被动运动的概率。

2) 洋流运动：当 $c(t)\geq 0.5$ 时，水母跟随洋流运动，带有随机扰动地向全局最优方向靠近，位置更新公式为：

$$X_i(t+1)=X_i(t)+rand(0,1)\times\left(X_{\mathrm{best}}-\beta\times rand(0,1)\times\mu\right) \tag{2}$$

其中，$\beta=3$ 为洋流系数，μ 为当前种群所有个体的均值，X_{best} 为当前全局最优解，$rand(0,1)$ 为[0, 1]之间的随机数，$X_i(t)$ 为第 i 个个体在第 t 代的位置。

3) 被动运动：当 $rand(0,1)>1-c(t)$ 时，水母在海洋中随机被动运动，公式为：

$$X_i(t+1)=X_i(t)+\gamma\times rand(0,1)\times(U_b-L_b) \tag{3}$$

其中，$\gamma=0.1$ 为被动运动系数，U_b 和 L_b 分别为优化变量的上界和下界。

4) 主动运动：当 $rand(0,1)\leq 1-c(t)$ 时，水母主动向其他个体靠近或远离，公式为：

$$X_i(t+1)=\begin{cases}X_i(t)+rand(0,1)\times\left(X_j(t)-X_i(t)\right) & f(X_i)\geq f(X_j)\\ X_i(t)+rand(0,1)\times\left(X_i(t)-X_j(t)\right) & f(X_i)<f(X_j)\end{cases} \tag{4}$$

其中，X_j 为随机选取的种群中另一个体，$f(\cdot)$ 为适应度函数，用于评价个体的优劣。

JSO 算法的流程为：初始化种群→计算适应度→更新全局最优→迭代执行三种运动行为→满足终止条件后输出全局最优解。

2.2. Otsu 多阈值分割原理

Otsu 算法[9]是一种经典的自适应阈值选取方法，其核心思想是通过选取最优阈值，将图像灰度级划分为多个区域，使区域间的类间方差最大化，区域内的类内方差最小化。

假设图像灰度级范围为[0, 255]，选取 k 个阈值 $t_1,t_2,\cdots,t_k$，其中，$0<t_1<t_2<\cdots<t_k<255$，将图像划分为 $k+1$ 个灰度区域 $C_0,C_1,\cdots,C_k$，其中 C_0 对应灰度级 $[0,t_1]$，C_1 对应灰度级 $[t_1+1,t_2]$，C_k 对应灰度级 $[t_k+1,255]$。

类间方差 σ_B^2 的计算公式为：

$$\sigma_B^2=\sum_{i=0}^{k}p_i\left(m_i-m_G\right)^2 \tag{5}$$

其中，p_i 为第 i 个区域的像素占比(即该区域像素数与图像总像素数的比值)，m_i 为第 i 个区域的灰度均值，m_G 为图像的全局灰度均值。

Otsu 多阈值分割的目标是寻找一组阈值 $\mathbf{t}=[t_1,t_2,\cdots,t_k]$，使类间方差 σ_B^2 最大化。由于本文采用最小化优化策略，因此将适应度函数设置为类间方差的负值，即：

$$f(\mathbf{t})=-\sigma_B^2=-\sum_{i=0}^{k}p_i\cdot\left(m_i-m_G\right)^2 \tag{6}$$

3. 改进水母优化算法

针对 JSO 算法的缺陷，本文引入自适应时间控制、精英引导策略以及硬边界约束三大改进策略，构建改进的水母优化算法，这些策略已经在粒子群优化算法(Particle Swarm Optimization, PSO)、鲸鱼优化算法(Whale Optimization Algorithm, WOA)，灰狼优化算法(Grey Wolf Optimizer, GWO)、萤火虫算法、海鸥优化算法等算法的改进中被证实是有效的[10]-[14]。本文结合水母在海洋中的觅食行为特点，具体改进如下。

3.1. 自适应时间控制

JSO 的时间控制系数采用线性衰减模式，难以平衡算法前期的全局探索能力和后期的局部开发能力。迭代初期，算法需要长时间、高强度的全局探索，但线性衰减会让系数过早下降，削弱探索能力，导致探索不足；而在迭代后期，算法需要快速收敛的局部开发，线性衰减后期下降过慢，搜索效率低、收敛精度差，导致开发不足。

本文针对 JSO 线性衰减模式探索后期“过早收敛、开发不足”的问题，设计了二次函数衰减模式。该模式在迭代前期下降平缓，保证了充分的全局探索；在迭代后期快速下降，强化局部开发能力，其参数设置与 JSO 的洋流运动、被动运动、主动运动切换机制直接耦合，实现自适应调整，提升收敛精度和速度。

自适应时间控制系数的公式为：

$$c(t)=c_0\cdot\left(1-\frac{t}{T}\right)^2 \tag{7}$$

其中，各参数含义与 JSO 一致。当迭代初期(t 较小时)，$c(t)$ 衰减较慢，保持较大值，使水母更易执行洋流运动，增强全局探索能力；当迭代后期(t 接近时 T)，$c(t)$ 衰减加快，取值较小，使水母更易执行被动运动和主动运动，增强局部开发能力。另外这里不考虑引入随机数是因为随机数的探索收益远小于其带来的稳定性损失，且改进公式通过非线性衰减已实现了更强的探索能力。

3.2. 精英引导策略

JSO 仅通过全局最优解引导种群更新，收敛精度有限。本文引入精英引导策略，选取种群中适应度最优的前 20%个体作为精英个体，融合全局最优解和精英个体均值，引导其他个体更新位置，提升算法的收敛精度和稳定性。精英引导策略并非简单引入外部精英个体，而是基于 JSO 算法中个体跟随洋流的行为模式，将种群中的优质个体作为动态洋流的核心，引导种群向高适应度区域迁移。

精英引导的位置更新公式为：

$$X_{\text{new}} = X_i + w_1 \cdot \left(X_{\text{best}} - X_i\right) + w_2 \cdot \left(\mu_{\text{elite}} - X_i\right) \tag{8}$$

其中，μ_{elite} 为精英个体的均值向量，$w_1 = 0.7 \cdot \left(1 - \frac{t}{T}\right)$ 为全局最优解的权重，迭代前期 w_1 较大，侧重全局探索；迭代后期 w_1 减小，侧重局部开发；$w_2 = 0.3 \cdot \frac{t}{T}$ 为精英个体均值的权重，迭代后期 w_2 增大，增强精英个体的引导作用。

该策略仅在水母执行主动运动后加入，既保证了种群的多样性，又能引导个体向更优方向更新，避免算法陷入局部最优。

3.3. 硬边界约束

JSO 采用 clip 函数实现边界约束，但未明确界定边界处理逻辑。本文将硬边界约束作为独立模块，明确采用 clip 函数将个体位置限制在解空间 $[L_b, U_b]$ 内，避免个体超出解空间导致适应度计算异常，保证算法的稳定性。

硬边界约束的公式为：

$$X_i = \text{clip}\left(X_i, L_b, U_b\right) \tag{9}$$

其中 $\text{clip}\left(x, a, b\right)$ 表示当 $x < a$ 时取 a，当 $x > b$ 时取 b，否则取 x。

3.4. IJS 算法流程

结合上述改进策略，IJS 算法的完整流程如下：

输入：优化问题维度 dim、解空间上下界 L_b / U_b、最大迭代次数 T、种群规模 N、适应度函数 $f(\cdot)$；

输出：全局最优解 X_{best}、全局最优适应度值 f_{best}。

1. 初始化算法参数：设置初始时间控制系数 $c_0 = 0.5$、洋流系数 $\beta = 3$、被动运动系数 $\gamma = 0.1$、精英比例 $\text{elite\_rate} = 0.2$、混沌系数 $\lambda = 4$，计算精英个体数量 $\text{elite\_size} = \lfloor N \times \text{elite\_rate} \rfloor$；

2. 混沌初始化种群：通过 Logistic 混沌映射生成混沌序列，映射到解空间，得到初始种群 $X = \{X_1, X_2, \cdots, X_N\}$；

3. 计算初始适应度：对种群中每个个体 X_i，计算其适应度 $f_i = f\left(X_i\right)$，记录全局最优解 $X_{\text{best}} = X_{\text{argmin}(f_i)}$ 和全局最优适应度 $f_{\text{best}} = \min\left(f_i\right)$；

4. 迭代优化（$t = 1, \cdots, T$）：计算自适应时间控制系数 $c(t) = c_0 \cdot \left(1 - \frac{t}{T}\right)^2$；计算当前种群均值 μ；

5. 选取精英个体：对适应度进行排序，选取前 elite_size 个个体作为精英个体，计算精英个体均值 μ_{elite}；

6. 个体位置更新：对每个个体 X_i，生成随机数 $rand$；若 $rand < c(t)$，执行洋流运动，更新位置 $X_i(t+1) = X_i(t) + rand \times \left(X_{\text{best}} - \beta \times rand \times \mu\right)$；否则，生成随机数 $rand1$；若 $rand1 < 0.5$：执行被动运动，更新位置 $X_i(t+1) = X_i(t) + \gamma \times rand1 \times \left(U_b - L_b\right)$；否则，执行主动运动，随机选取个体 X_j，根据适应度大小更新位

置，再加入精英引导策略调整位置；

7. 硬边界约束：对更新后的个体位置进行边界处理 $X_i = \text{clip}(X_i, L_b, U_b)$；

8. 适应度更新：计算更新后个体的适应度 $f_{\text{new}} = f(X_i)$，若 $f_{\text{new}} < f_i$，则更新个体位置和适应度；若 $f_{\text{new}} < f_{\text{best}}$，则更新全局最优解和全局最优适应度；

9. 迭代终止：当迭代次数达到 T 时，输出全局最优解 X_{best} 和全局最优适应度 f_{best}。

4. 实验设计与结果分析

4.1. 实验参数设置

为保证实验的公平性和可比性，所有算法的参数设置保持一致(除算法自身特有参数外)，具体参数如下：

种群规模 $N = 30$，最大迭代次数 $T = 500$；独立运行次数为 30 次；

IJS 特有参数：精英比例 0.2；

PSO 特有参数：惯性权重 $w = 0.7$，加速系数 $c_1 = c_2 = 1.49445$；

WOA 特有参数：螺旋形状参数 $b = 1$；

其他算法参数均采用原始论文推荐参数，与源代码中设置一致。

4.2. 函数寻优实验

4.2.1. 测试函数选取

选取 6 个标准测试函数，覆盖低维/高维、单峰/多峰、含噪声/无噪声等不同特性，全面验证 IJS 算法的寻优性能。测试函数的详细信息如表 1 所示。

Table 1. Benchmark functions
表 1. 标准测试函数

函数名称	缩写	维度	搜索范围	理论最优值
Stepint 函数	F1	5	[−5.12, 5.12]	0
Step 函数	F2	30	[−100, 100]	0
Sphere 函数	F3	30	[−100, 100]	0
SumSquares 函数	F4	30	[−10, 10]	0
Quartic 函数	F5	30	[−1.28, 1.28]	0
Beale 函数	F6	2	[−4.5, 4.5]	0

4.2.2. 实验指标

选取 4 个核心指标评价算法的寻优性能，分别为：

最优值：30 次独立运行中得到的最小适应度值，越接近理论最优值越好；

均值：30 次独立运行最优值的平均值，反映算法的整体寻优水平，越接近理论最优值越好；

标准差：30 次独立运行最优值的标准差，反映算法的稳定性，越小越好；

时间开销：单次运行的平均时间(单位：s)，反映算法的效率，越小越好。

4.2.3. 实验结果与分析

为了验证 IJS 算法的有效性，本文将其与 JSO，GWO [15]，PSO [16]，WOA [17]，进行对比，各算

法在 6 个标准测试函数上的 30 次独立运行统计结果如下表 2 所示。

Table 2. Optimization results of different algorithms on benchmark functions
表 2. 标准测试函数下不同算法的优化结果

测试函数	算法	最优值	均值	标准差	时间开销
F1	IJS	0	0	0	0.1658
	JSO	0	0	0	0.1184
	GWO	0	0	0	0.2935
	PSO	0	0	0	0.1554
	WOA	0	0	0	0.3101
F2	IJS	0	0	0	0.6107
	JSO	0	0	0	0.4329
	GWO	0	0	0	0.877
	PSO	0	12.73333333	16.04355184	0.5128
	WOA	0	0	0	0.6351
F3	IJS	1.931E−71	2.01574E−66	6.76595E−66	0.6237
	JSO	6.62007E−26	7.66199E−24	9.34378E−24	0.421
	GWO	2.65784E−32	1.02243E−30	2.48239E−30	0.9019
	PSO	2.78096E−09	333.3333341	1795.054936	0.5015
	WOA	2.00093E−51	2.0164E−47	6.57711E−47	0.6309
F4	IJS	3.21737E−72	8.13581E−68	2.2685E−67	0.6699
	JSO	5.95276E−26	2.20283E−24	4.89248E−24	0.4852
	GWO	3.42912E−33	1.46022E−31	5.15333E−31	0.9735
	PSO	6.68913E−10	146.6666668	209.3375795	0.5534
	WOA	5.0696E−54	3.90445E−49	1.43461E−48	0.6945
F5	IJS	6.12502E−05	0.000340224	0.000267215	0.6821
	JSO	0.001185506	0.002529527	0.00095402	0.5022
	GWO	0.000980617	0.002424646	0.001090418	1.0138
	PSO	0.020207312	0.141603436	0.479414065	0.6036
	WOA	0.002274575	0.010177088	0.005988558	0.7173
F6	IJS	0	1.13326E−06	4.14105E−06	0.4841
	JSO	0	5.41919E−26	2.56673E−25	0.294
	GWO	5.6062E−09	0.025402511	0.136795657	0.7844
	PSO	0	0.025402322	0.136795689	0.3708
	WOA	0	0	0	0.4906

由表中结果可知，在 6 个测试函数上，各算法的性能差异较为明显。对于 F1 和 F2 这类相对简单的函数，IJS、JSO、GWO 和 WOA 基本均能够稳定获得全局最优解，说明各算法在低难度问题上均具备较

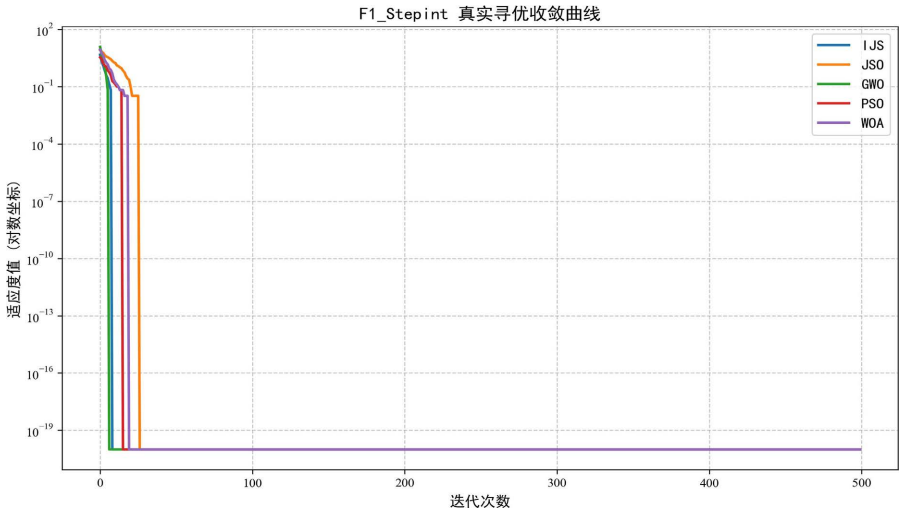
好的搜索能力，此时差异主要体现在运行效率上，其中 JSO 的时间开销最小，表现出较高的计算效率。相比之下，PSO 在 F2 上已出现一定波动，其均值和标准差明显大于其余算法，说明其稳定性相对不足。总体来看，在简单测试函数上，各算法精度差异不大，而 JSO 在速度方面更具优势。

当问题复杂度提高后，IJS 的优势开始显著体现。在 F3、F4 和 F5 三个函数上，IJS 的最优值、均值和标准差均优于其他对比算法，表现出更高的寻优精度和更强的稳定性。尤其相较于原始 JSO，IJS 在 F3 和 F4 上的均值与标准差均实现了数量级上的下降，表明所引入的自适应时间控制、精英引导策略和硬边界约束能够有效增强算法的全局探索能力与局部开发能力，显著缓解 JSO 易早熟收敛、后期开发能力不足的问题。与此同时，WOA 在 F3、F4 及 F6 上也表现出一定竞争力，GWO 整体性能居中，而 PSO 在多个复杂函数上均值和标准差偏大，表现出较明显的早熟收敛倾向和鲁棒性不足的问题。

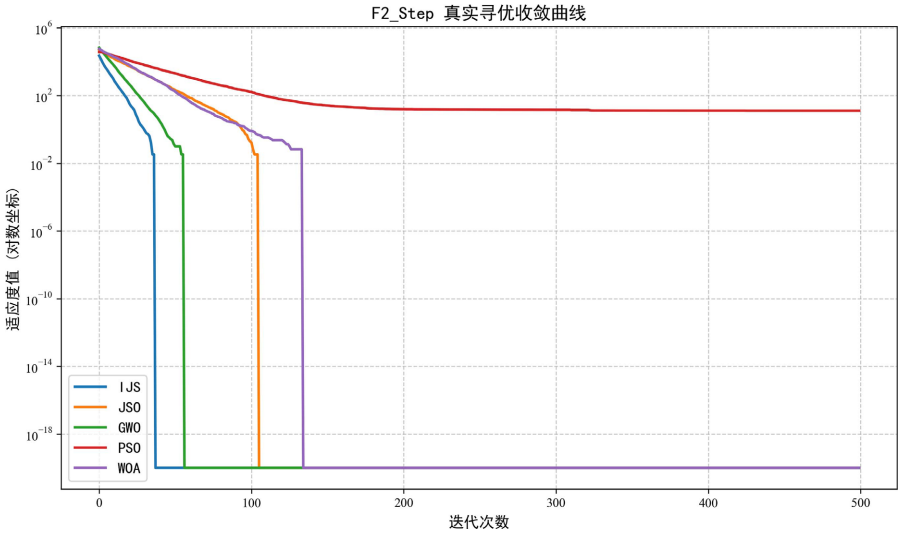
另外，IJS 虽然在时间开销上高于原始 JSO，但其平均耗时仍处于可接受范围，且明显优于部分精度相近但计算代价更高的算法。特别是在复杂函数优化中，IJS 通过适度增加时间成本，换取了更显著的精度提升和更稳定的求解结果，体现出更优的整体平衡性。值得注意的是，在 F6 上 IJS 虽能取得最优值，但均值和标准差不及 WOA 与 JSO。进一步分析发现，F6 函数的图形为连续但带局部振荡的二次曲面，其全局最优位置位于原点，梯度信息相对简单。IJS 在迭代后期引入的精英引导策略，会使种群过度向当前最优个体聚集，在这类梯度信息单一的问题上容易因过度开发而陷入局部最优，同时精英个体的引导作用也放大了随机扰动，导致算法在 F6 上的稳定性下降。此外，本文设计的二次衰减自适应时间控制策略在 F6 的快速收敛阶段未能及时切换至探索模式，加剧了算法的早熟收敛倾向。后续将针对此类问题优化精英个体的选择比例与衰减系数，以进一步提升算法的鲁棒性。总体而言，IJS 在精度、稳定性和效率之间实现了较好的协调统一，综合性能优于 JSO 及其余对比算法，验证了所提改进策略的有效性与实用性。

图 1 给出了各算法针对不同标准测试函数上的收敛曲线。

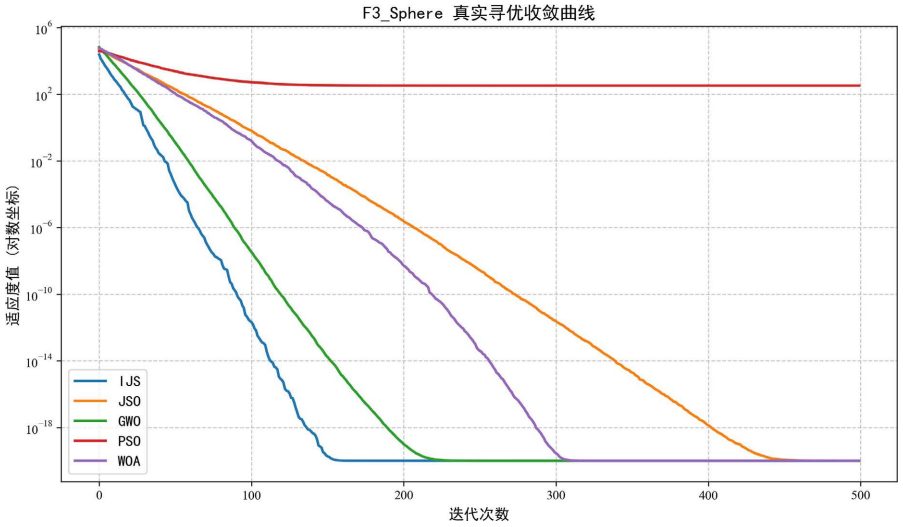
由图 1 中各测试函数的真实寻优收敛曲线可以看出，纵轴采用对数坐标后，不同算法在收敛速度、收敛精度以及后期稳定性方面的差异表现得更为明显。总体而言，IJS 在大多数函数上表现出更快的前期下降速度和更低的最终适应度值，说明其在全局探索与局部开发之间具有较好的平衡能力。尤其在 F2、F3、F4 和 F5 上，IJS 均能够在较少迭代次数内迅速降低适应度，并率先进入稳定平台区，表现出较强的快速寻优能力。相比之下，JSO 虽然整体收敛过程较为平稳，但在多数函数上下降速度偏慢，往往需要



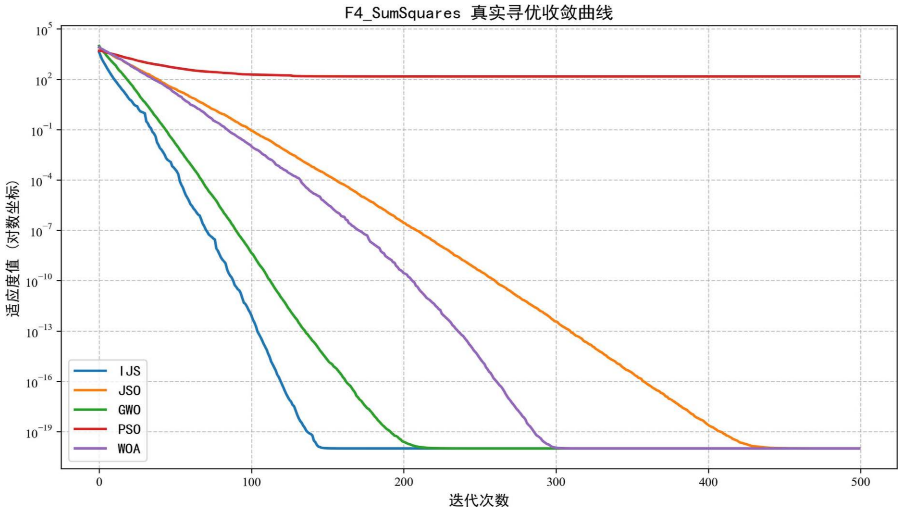
(a)


F2_Step 真实寻优收敛曲线
适应度值（对数坐标）
迭代次数
IJS
JSO
GWO
PSO
WOA

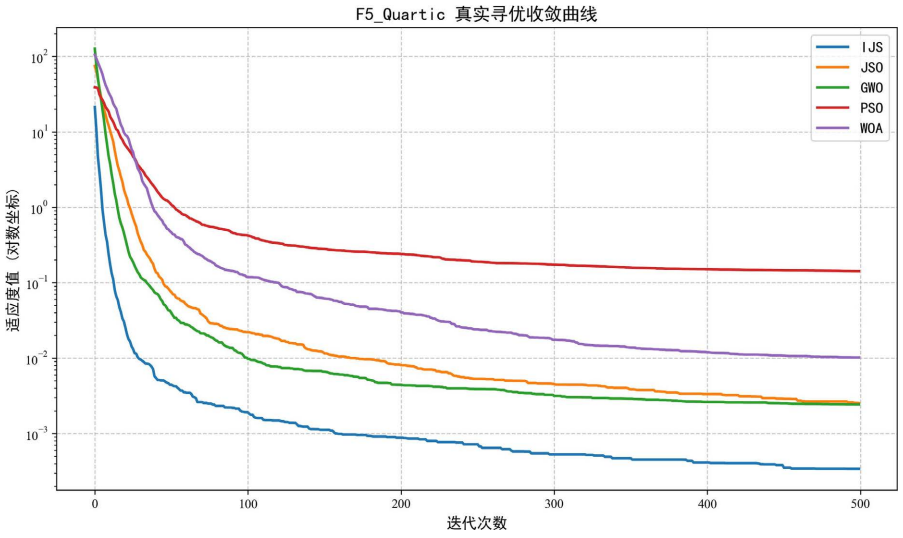
(b)


F3_Sphere 真实寻优收敛曲线
适应度值（对数坐标）
迭代次数
IJS
JSO
GWO
PSO
WOA

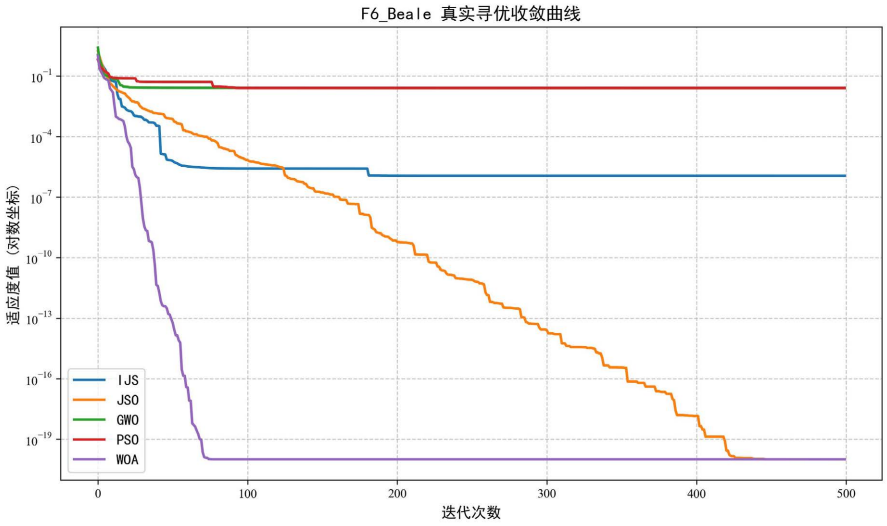
(c)


F4_SumSquares 真实寻优收敛曲线
适应度值（对数坐标）
迭代次数
IJS
JSO
GWO
PSO
WOA

(d)



(e)



(f)

Figure 1. Convergence curves of different benchmark functions
图 1. 不同基准函数的收敛曲线

更多迭代次数才能逼近最优区域；GWO 和 WOA 在部分函数上具有一定竞争力，但综合表现不及 IJS；PSO 则在多个函数上出现明显的平台停滞现象，表现出较强的早熟收敛倾向。

5. 基于 IJS 的图像多阈值分割实验

本文将 IJS 应用于多尺度图像分割，对比方法包括 Otsu 和 JSO，阈值维度设置为 2、4、6、8、10 维。评价指标包括特征相似性(Feature Similarity, FSIM)、结构相似性(Structural Similarity Index, SSIM)和峰值信噪比(Peak Signal-to-Noise Ratio, PSNR)。为了进一步显示改进算法的有效性，在此实验中最大迭代次数设置为 20 次。不同方法的实验结果如图 2 和表 3。

由表 3 中各算法的 FSIM、SSIM 和 PSNR 三项评价指标结果可以看出，不同算法在图像分割的特征保留、结构一致性以及抗噪声能力方面的差异表现得较为明显。总体而言，IJS 系列算法(IJS-2 至 IJS-10)

整体表现优于原始 Otsu 算法和 JSO 系列算法(JSO-2 至 JSO-10)，且随着阈值维度从 2 增加到 10，IJS 系列算法的各项指标整体呈上升趋势，说明改进策略能够有效提升图像分割质量；JSO 系列算法与对应参数的 IJS 算法表现相近，但在部分参数下略逊于 IJS 算法，整体分割效果优于 Otsu 算法但不及 IJS 系列的最优表现。Otsu 算法作为传统阈值分割方法，各项指标均为最低，说明其在复杂图像分割中难以兼顾特征保留与结构一致性，抗噪声能力较弱。

Table 3. Evaluation metrics of different image segmentation methods
表 3. 不同图像分割方法评价指标

方法	FSIM	SSIM	PSNR
Otsu	0.2195	0.607	12.8581
IJS-2	0.2985	0.6912	17.1078
JSO-2	0.2985	0.6912	17.1078
IJS-3	0.3623	0.7316	20.0552
JSO-3	0.3567	0.73	20.6466
IJS-4	0.4111	0.7797	22.5192
JSO-4	0.4174	0.7784	22.4782
IJS-6	0.4928	0.8369	23.6011
JSO-6	0.4769	0.8303	23.7936
IJS-8	0.5427	0.8647	22.7772
JSO-8	0.5126	0.8502	21.1062
IJS-10	0.593	0.8866	27.5948
JSO-10	0.549	0.8701	24.036

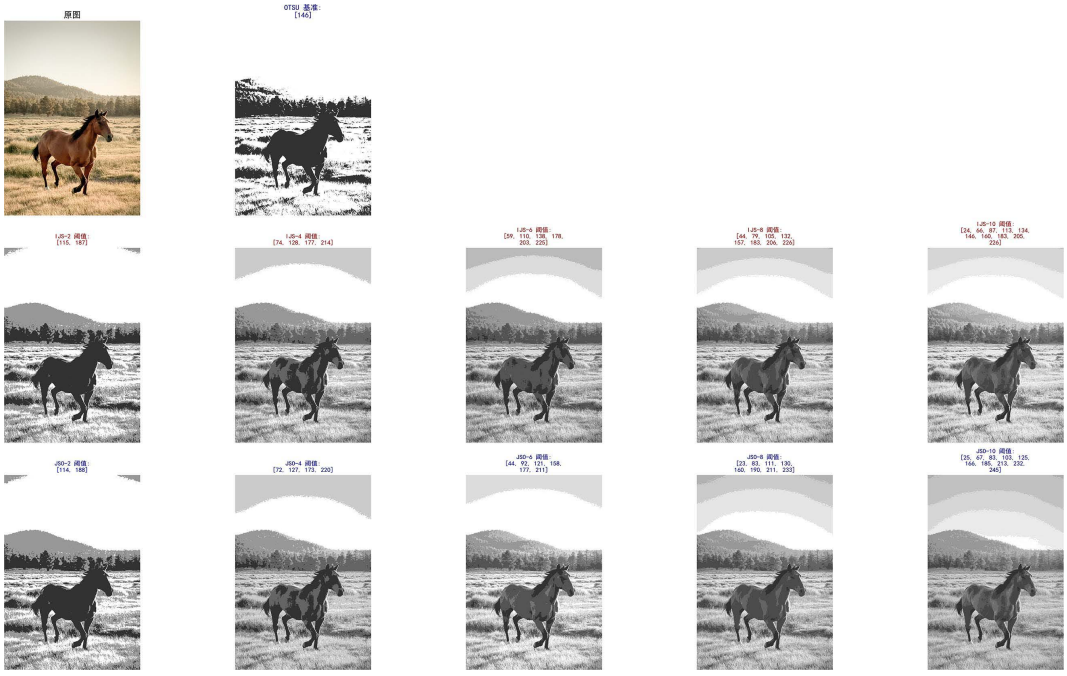


Figure 2. Segmentation results with different thresholds
图 2. 不同阈值下分割效果图

由图 2 结果可知，传统 Otsu 单阈值法仅能实现二值化分割，细节丢失严重，无法还原图像的灰度层次与纹理信息；随着阈值维度从 2 提升至 10，IJS 与 JSO 算法的分割效果均持续优化，灰度层次逐步丰富，细节还原度不断提升。在相同维度下，IJS 算法的分割效果显著优于 JSO 算法，其灰度分层更均匀、细节保留更完整、边缘过渡更平滑，验证了自适应时间控制系数、精英引导等改进策略的有效性。当阈值维度达到 10 时，IJS 算法的分割结果最接近原图，细节还原度最高。综合来看，IJS 算法在多阈值图像分割中具有明显的性能优势，能够有效解决传统 Otsu 法的缺陷，为复杂图像的高精度分割提供了可靠方案。

不同阈值下的收敛曲线如图 3 所示。

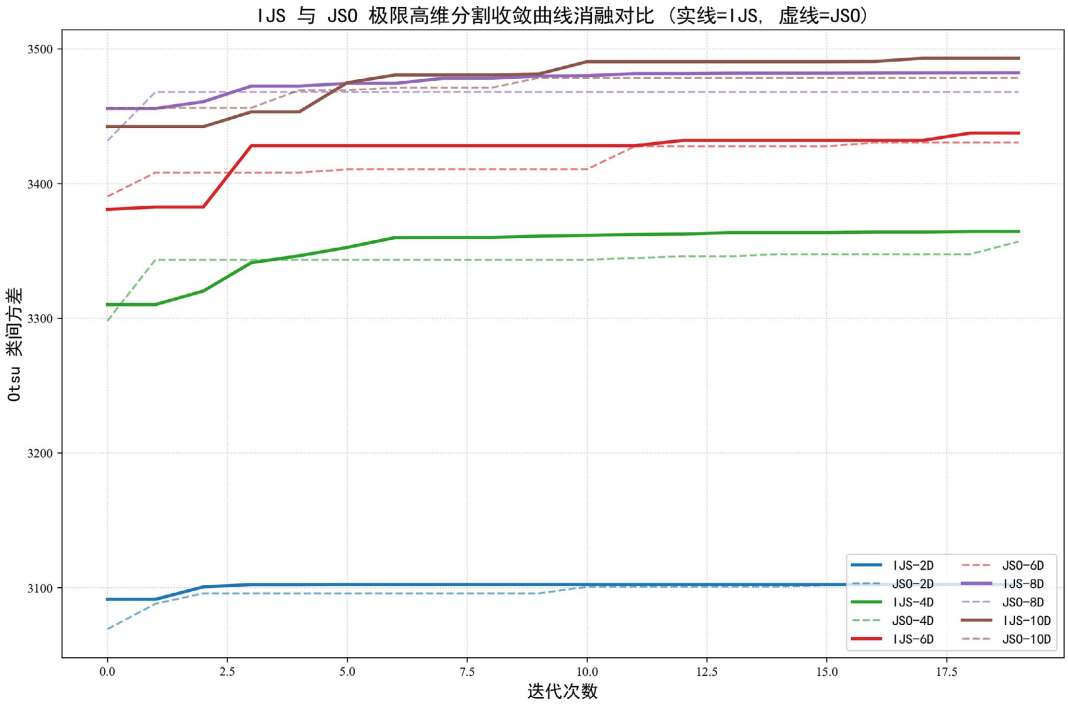


Figure 3. Convergence curves under different thresholds
图 3. 不同阈值下的收敛曲线

由 IJS 与 JSO 在极限高维分割的收敛曲线可以看出，纵轴 Otsu 类间方差越高代表分割阈值的类间区分度越优、分割质量越好，不同算法在收敛速度、收敛精度以及后期稳定性方面的差异表现得极为明显。总体而言，IJS 在所有维度上均表现出更快的前期收敛速度和更高的最终类间方差值，说明其在全局探索与局部开发之间具有更优的平衡能力。尤其在阈值维度为 6、8、10 时的高维场景下，IJS 均能够在较少迭代次数内迅速提升类间方差，并率先进入稳定平台区，表现出极强的快速寻优与高精度收敛能力。相比之下，JSO 虽然整体收敛过程较为平稳，但在所有维度上的收敛速度偏慢，最终收敛精度均低于 IJS，且在高维场景下的性能差距被进一步放大，说明其在高维多阈值寻优中易陷入局部最优，后期精细开发能力不足。

6. 结论

本文针对 JSO 存在的线性时间系数衰减不合理、精英信息利用不足、高维易早熟收敛等缺陷，提出改进水母优化算法。通过构建自适应二次时间控制策略、精英个体引导机制，实现了算法探索与开发能

力的动态平衡，有效提升了收敛速度、寻优精度与运行稳定性。

在 6 个标准测试函数上，IJS 的整体表现优于 JSO、GWO、PSO、WOA 算法，尤其在高维复杂函数上优势更为明显，收敛速度较 JSO 提升显著，同时时间开销保持在合理范围。收敛曲线对比显示，IJS 能够在更少迭代次数内快速逼近最优值，并稳定进入收敛平台，有效克服了 JSO 后期搜索缓慢、易陷入局部最优的问题。

在图像多阈值分割应用中，IJS 基于 Otsu 类间方差最大化准则实现多维阈值自动寻优，分割结果在边缘清晰度、灰度层次均匀性与细节保留度上均优于传统 Otsu 算法与 JSO 算法。FSIM、SSIM、PSNR 指标表明，随着阈值维度升高，IJS 分割质量持续提升，在高维分割中稳定性更强、精度更高，能够更好地还原图像结构与纹理信息。

基金项目

山西省基础研究计划项目(202303021212255, 202303021212164)，山西省高等学校科技创新项目(2022L405)。

参考文献

[1] Tang, J., Liu, G. and Pan, Q. (2021) A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends. *IEEE/CAA Journal of Automatica Sinica*, **8**, 1627-1643. https://doi.org/10.1109/jas.2021.1004129

[2] Chou, J.S. and Truong, D.N. (2021) A Novel Metaheuristic Optimizer Inspired by Behavior of Jellyfish in Ocean. *Applied Mathematics and Computation*, **389**, Article 125535. https://doi.org/10.1016/j.amc.2020.125535

[3] Yildizdan, G. (2022) MJS: A Modified Artificial Jellyfish Search Algorithm for Continuous Optimization Problems. *Neural Computing and Applications*, **35**, 3483-3519. https://doi.org/10.1007/s00521-022-07842-w

[4] 肖辉辉, 段艳明. 求解 UCAV 航路规划问题的改进水母算法[J/OL]. 计算机技术与发展, 2026: 1-10. https://doi.org/10.20165/j.cnki.ISSN1673-629X.2026.0040, 2026-05-27.

[5] Guo, S.X., Chen, M.Z. and Pang, W. (2023) Path Planning for Autonomous Underwater Vehicles Based on an Improved Artificial Jellyfish Search Algorithm in Multi-Obstacle Ocean Current Environment. *IEEE Access*, **11**, 31010-31023. https://doi.org/10.1109/access.2023.3257025

[6] 陶鑫杰, 莫愿斌. 融合多策略人工水母算法及工程应用研究[J]. 现代电子技术, 2023, 46(7): 85-90.

[7] Srikanth, R. and Bikshalu, K. (2021) Multilevel Thresholding Image Segmentation Based on Energy Curve with Harmony Search Algorithm. *Ain Shams Engineering Journal*, **12**, 1-20. https://doi.org/10.1016/j.asej.2020.09.003

[8] Liao, P.S., Chen, T.S. and Chung, P.C. (2001) A Fast Algorithm for Multilevel Thresholding. *Journal of Information Science and Engineering*, **17**, 713-727.

[9] Otsu, N. (1979) A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**, 62-66. https://doi.org/10.1109/tsmc.1979.431007

[10] 秦飞, 陈丹凤. 基于自适应粒子群优化算法的液压系统可靠性优化[J]. 白城师范学院学报, 2026, 40(2): 41-51.

[11] 米强, 蒋强. 基于多目标萤火虫算法的机械臂轨迹规划[J]. 通信与信息技术, 2026(2): 41-46.

[12] 赵晏淇, 李伯群, 张春宇, 等. 基于改进海鸥优化算法的热连轧板带轧制规程优化[J]. 塑性工程学报, 2024, 31(11): 97-104.

[13] 巢渊, 徐魏, 刘文汇, 等. 基于改进灰狼优化算法的 QFN 芯片图像多阈值分割方法[J]. 光学精密工程, 2024, 32(6): 930-944.

[14] 黄辉先, 张广炎, 陈思溢, 等. 基于混沌权重和精英引导的鲸鱼优化算法[J]. 传感器与微系统, 2020, 39(5): 113-116.

[15] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. 1995 *International Conference on Neural Networks*, Perth, 27 November-1 December 1995, 1942-1948. https://doi.org/10.1109/icnn.1995.488968

[16] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[17] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008