

# 一种改进型多元宇宙优化算法

李志明<sup>1</sup>, 谢永盛<sup>2\*</sup>, 周加全<sup>1</sup>, 黄秀芳<sup>1</sup>, 吴豆豆<sup>1</sup>

<sup>1</sup>广西科技师范学院数学与计算机工程学院, 广西 来宾

<sup>2</sup>广西科技师范学院人工智能学院, 广西 来宾

收稿日期: 2025年12月3日; 录用日期: 2026年1月2日; 发布日期: 2026年1月9日

## 摘要

针对多元宇宙优化算法在优化问题求解中存在的不足, 本文结合传统的单纯性法和Lévy飞行策略, 提出一种改进的多元宇宙优化算法。该算法通过融合两种策略, 显著提升了原算法的求解精度并加快了收敛速度。经11个标准测试函数验证, 新算法在函数优化中展现出较强优势, 测试结果更接近理论最优值。将其应用到工程优化问题时, 亦取得了较好效果。

## 关键词

最优化, 多元宇宙优化, 单纯性, Lévy飞行, 函数优化, 焊接梁设计

# An Improved Multi-Verse Optimizer

Zhiming Li<sup>1</sup>, Yongsheng Xie<sup>2\*</sup>, Jiaquan Zhou<sup>1</sup>, Xiufang Huang<sup>1</sup>, Doudou Wu<sup>1</sup>

<sup>1</sup>College of Mathematics and Computer Engineering, Guangxi Science & Technology Normal University, Laibin Guangxi

<sup>2</sup>College of Artificial Intelligence, Guangxi Science & Technology Normal University, Laibin Guangxi

Received: December 3, 2025; accepted: January 2, 2026; published: January 9, 2026

## Abstract

To address the deficiencies of the Multi-Verse Optimizer in solving optimization problems, this paper proposes an improved Multi-Verse Optimizer by combining the traditional simplex method with the Lévy flight strategy. By integrating these two strategies, the proposed algorithm significantly enhances the solution accuracy of the original MVO and accelerates its convergence speed. Verified by 11 standard test functions, the new algorithm shows strong advantages in function optimization, with test results being closer to the theoretical optimal values. When applied to engineering optimization problems, it also achieves favorable effects.

\*通讯作者。

文章引用: 李志明, 谢永盛, 周加全, 黄秀芳, 吴豆豆. 一种改进型多元宇宙优化算法[J]. 计算机科学与应用, 2026, 16(1): 28-43. DOI: 10.12677/csa.2026.161004

## Keywords

Optimization, Multi-Verse Optimizer, Simplex Method, Lévy Flights, Function Optimization, Welded Beam Design

Copyright © 2026 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

元启发式算法的思想多源于人类对生物、物理、社会等领域现象的长期观察，通过对这些现象的深入分析总结出自然规律，是人类智慧的结晶。与传统优化方法相比，元启发式算法凭借能简单高效地解决复杂问题的特性，在工程领域获得了广泛认可与应用[1]。蝙蝠算法(BA) [2]、布谷鸟算法(CS) [3]、花授粉算法(FPA) [4]等元启发式算法，将待优化问题的可能解视为解空间，具有不受搜索空间连续性或可微性限制等显著优点，因此已在诸多领域得到广泛应用。

学者 Mirjalili 受宇宙大爆炸理论启发而提出多元宇宙优化算法(Multi-Verse Optimizer, MVO) [5]，该算法中宇宙大爆炸表明宇宙有一个膨胀率，白洞具有较高膨胀率，黑洞具有较低膨胀率，而物质通过虫洞从白洞向黑洞移动，多元宇宙优化算法便是模拟了此现象。无论宇宙中物质的膨胀率大小如何，其中的物质都会通过虫洞随机移动，到达最佳位置。算法中每个解都相当于一个宇宙，解中的每个变量都是该宇宙中的物质。此外，每个解分配有膨胀率，该膨胀率与解的相应适应性函数值成比例。由于多元宇宙优化算法的全局寻优能力较好，需要调节的参数较少，性能较为稳定并且易于实现，已经成功在文本聚类[6]、桁架设计[7]、大规模离散时间 - 成本权衡[8]等实际问题中得到应用。

## 2. 多元宇宙优化算法

### 2.1. MVO 算法

多元宇宙算法利用白洞和黑洞的相互作用进行搜索空间的探索，虫洞作为转移物体的媒介在算法中优化开发搜索空间。每个宇宙都是问题的一个解，宇宙中的物体代表解的变量，宇宙膨胀率代表解的适应度值。迭代时，根据宇宙膨胀率进行排序，依靠轮盘赌机制来产生白洞。

设宇宙矩阵用  $U$  表示：

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{bmatrix} \quad (1)$$

其中， $d$  为变量个数； $n$  为宇宙数(候选解)。

$$x_i^j = \begin{cases} x_k^j & r_1 < NI(U_i) \\ x_i^j & r_2 \geq NI(U_i) \end{cases} \quad (2)$$

其中  $x_i^j$  表示第  $i$  个宇宙的第  $j$  个变量； $U_i$  表示第  $i$  个宇宙， $NI(U_i)$  是第  $i$  个宇宙的标准膨胀率， $r_1$  是区间[0,1]中的随机数， $x_k^j$  表示第  $k$  个宇宙的第  $j$  个变量(由轮盘赌选择机制选择)。

为了增加提高膨胀率的概率，使用到以下公式：

$$x_i^j = \begin{cases} X_j + TDR \times ((ub_j - lb_j) \times r4 + lb_j) & r3 < 0.5 \\ X_j - TDR \times ((ub_j - lb_j) \times r4 + lb_j) & r3 \geq 0.5 \end{cases} \quad \begin{matrix} r2 < WEP \\ r2 \geq WEP \end{matrix} \quad (3)$$

其中  $X_j$  表示目前最优宇宙的第  $j$  个变量,  $TDR$  和  $WEP$  是两个系数,  $lb_j$  表示第  $j$  个变量的最大值,  $ub_j$  是第  $j$  个变量的最小值,  $x_i^j$  表示第  $i$  个宇宙的第  $j$  个变量,  $r2, r3, r4$  是  $[0,1]$  中的随机数。另外, 虫洞存在概率  $WEP$  和旅行距离率  $TDR$  是多元宇宙优化算法的两个重要参数。

$$WEP = \min + l \times \left( \frac{\max - \min}{L} \right) \quad (4)$$

公式中,  $\min$  为  $WEP$  最小值(设为 0.2),  $\max$  为  $WEP$  最大值(设为 1),  $l$  为当前迭代次数;  $L$  为最大迭代次数。

$$TDR = 1 - \left( \frac{l}{L} \right)^{\frac{1}{p}} \quad (5)$$

$p$  (设为 6) 变化的探索速度,  $p$  值越高, 局部探索速度越快, 用时越短。两个参数随时间变化曲线如图 1 所示。

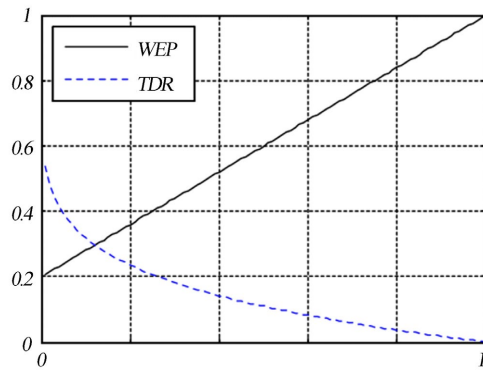


Figure 1. Wormhole existence probability versus travelling distance rate  
图 1. WEP 和 TDR 变化曲线

## 2.2. 单纯性

图 2 为单纯形法示意图[9], 其具体实施步骤如下。

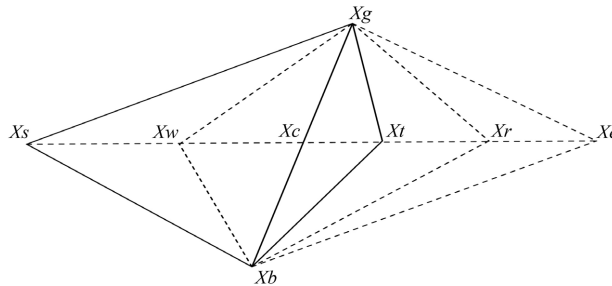


Figure 2. Simplex method  
图 2. 单纯形法示意图

**Step 1:** 从所有解中找到全局最优解  $F_1 \sim F_6$  以及次优解  $F_1 \sim F_6$ ，设  $F_1 \sim F_6$  是较差解，其对应的适应度值分别记为  $F_1 \sim F_6$ ， $F_1 \sim F_6$  和  $F_1 \sim F_6$ ；

**Step 2:** 利用公式(6)计算出  $F_1 \sim F_6$  和  $F_1 \sim F_6$  的中间点  $F_1 \sim F_6$ ；

$$F_1 \sim F_6 \quad (6)$$

**Step 3:** 使用公式(7)执行反射操作，得到反射点  $F_1 \sim F_6$ 。其中  $F_1 \sim F_6$  为反射率，通常设置成 1；

$$F_1 \sim F_6 \quad (7)$$

**Step 4:** 比较反射点和全局最优点的适应度值  $F_1 \sim F_6$  和  $F_1 \sim F_6$ 。如果  $F_1 \sim F_6$ ，则通过公式(8)进行扩展操作：

$$F_1 \sim F_6 \quad (8)$$

$F_1 \sim F_6$  是扩充常数。拿扩展点和最优点的适应度值作比较，如果  $F_1 \sim F_6$ ，则用点  $F_1 \sim F_6$  代替  $F_1 \sim F_6$ 。否则用  $F_1 \sim F_6$  代替  $F_1 \sim F_6$ 。

**Step 5:** 将  $F_1 \sim F_6$  和  $F_1 \sim F_6$  的适应度值作比较。如果  $F_1 \sim F_6$ ，则通过公式(9)进行压缩操作：

$$F_1 \sim F_6 \quad (9)$$

其中  $F_1 \sim F_6$  为压缩系数，通常设置成 0.5。然后比较压缩点的适应度值  $F_1 \sim F_6$  和  $F_1 \sim F_6$ ，如果  $F_1 \sim F_6$ ，那么用压缩点  $F_1 \sim F_6$  代替  $F_1 \sim F_6$ 。否则用反射点  $F_1 \sim F_6$  代替  $F_1 \sim F_6$ 。

**Step 6:** 如果  $F_1 \sim F_6$ ，则由式(10)可得到收缩点  $F_1 \sim F_6$ ，计算其适应度值  $F_1 \sim F_6$

$$F_1 \sim F_6 \quad (10)$$

其中  $F_1 \sim F_6$  为收缩率，然后比较  $F_1 \sim F_6$  与  $F_1 \sim F_6$ 。若  $F_1 \sim F_6$ ，那么就用收缩点  $F_1 \sim F_6$  代替  $F_1 \sim F_6$ 。否则依旧用反射点  $F_1 \sim F_6$  代替  $F_1 \sim F_6$ 。

### 2.3. Lévy 飞行

Lévy 飞行是由法国数学家 Lévy 提出的一种随机游走模式，其特征表现为短距离移动的同时伴随偶尔的长距离跳跃。该特性能够有效避免系统陷入局部最优。Lévy 飞行可在不确定环境中最大化搜索效率，将其与多元宇宙优化算法相结合时，能够扩大算法的搜索范围、增加种群多样性，帮助算法更易跳出局部最优的停滞状态。在自然界中，一些动物为在不确定环境中高效觅食，采用的最优搜索策略正是 Lévy 飞行；大量研究也证实，许多动物的移动行为轨迹与 Lévy 飞行特征高度吻合。目前，Lévy 飞行已被成功应用于优化领域，相关研究结果表明其应用效果令人满意[10]。

### 2.4. 基于单纯性和 Lévy 飞行轨迹的多元宇宙优化算法

本文在多元宇宙优化算法的基础上，引入单纯形法与 Lévy 飞行策略，提出一种基于单纯形和 Lévy 飞行的多元宇宙优化算法(LSMVO)。其中，加入单纯形法的作用是对每次迭代的最优值进行扰动，而引入 Lévy 飞行的主要作用是避免算法陷入局部最优。通过在每次位置更新后融入单纯形法操作与 Lévy 随机位置移动，算法的搜索能力得到有效增强，整体性能也实现了较大提升。

## 3. 仿真实验

### 3.1. 实验环境与参数设置

为了验证 LSMVO 算法的有效性与可行性，文中采用 11 个基准测试函数来验证其性能。在此次仿真实验中，用到了蝙蝠算法(BA)、布谷鸟算法(CS)、花授粉算法(FPA)。其中 MVO 和 LSMVO 算法中种群

规模为 30，最大迭代次数为 500 次，函数维数设置为 30 维，最后采用独立运行 20 次最优值的平均值作为测试结果。对于 FPA，参数设置为  $p = 0.8$ ，参数研究表明， $p = 0.8$  对于大多数应用具有较好的效果；对于 CS，巢数设定为  $N=30$ ，外来蛋的发现率设定为  $p=0.25$ ；对于 BA，脉冲频率范围设置为  $Q=\{0,2\}$ ，最大响度设置为  $A=0.5$ ，最大脉冲发射设置为  $R=0.5$ ，响度衰减系数设定为  $C=0.95$ ，脉冲发射增加系数设定为  $C=0.05$ 。

### 3.2. 测试函数

在本次仿真实验选取的基准测试函数中， $F_1 \sim F_6$  为单峰函数， $F_7 \sim F_{11}$  为多峰函数。由于单峰函数只有一个全局最优值，因此适合用于衡量算法的开发能力。与单峰函数不同，多峰函数有大量的局部最优值，其对检验算法的探测能力及避免陷入局部最优具有重要作用。实验中用到的 11 个测试函数如表 1 所示：

**Table 1.** Benchmark test functions  
**表 1.** 测试函数

Benchmark Test Functions	D	Range	Optimum
$F_1 = \sum_{i=1}^D x_i^2$	30	$[-100,100]$	0
$F_2 = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	30	$[-10,10]$	0
$F_3 = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)$	30	$[-100,100]$	0
$F_4 = \max \{  x_i , 1 \leq i \leq D \}$	30	$[-100,100]$	0
$F_5 = \sum_{i=1}^D \left[ 100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2 \right]$	30	$[-30,30]$	0
$F_6 = \sum_{i=1}^D [x_i + 0.5]$	30	$[-100,100]$	0
$F_7 = \sum_{i=1}^D [x_i^2 - 10 \cos 2\pi x_i + 10]$	30	$[-5.12, 5.12]$	0
$F_8 = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$	30	$[-32,32]$	0
$F_9 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	30	$[-600,600]$	0
$F_{10} = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \left[ 1 + \sin^2(\pi y_{y_{i+1}}) \right] + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i, -z)^m & x_i < -a \end{cases}$	30	$[-50,50]$	0

续表

$$F_{11} = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^D (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right. \\ \left. + \sum_{i=1}^D u(x_i, 5, 100, 4) \right\} \quad 30 \quad [-50, 50] \quad 0$$

### 3.3. 实验结果

单峰函数实验结果如表 2 所示, 多峰函数实验结果如表 3 所示。

**Table 2.** Results of unimodal benchmark functions

**表 2.** 单峰函数实验结果

函数	结果	BA	CS	FPA	MVO	LSMVO
<b>F1</b>	最优值	2.8900E+04	1.1628E-01	4.4303E-01	6.0955E-01	<b>2.4779E-147</b>
	最差值	5.0388E+04	3.2955E-01	1.0333E+00	2.0232E+00	<b>2.6827E-74</b>
	平均值	4.1052E+04	1.8779E-01	7.1094E-01	1.3094E+00	<b>1.3413E-75</b>
	标准差	5.6270E+03	5.7876E-02	1.3053E-01	4.1312E-01	<b>5.9986E-75</b>
<b>F2</b>	最优值	1.6267E+04	2.1425E+00	2.5991E+00	4.5847E-01	<b>6.9593E-85</b>
	最差值	2.4811E+08	7.5428E+00	5.0776E+00	1.1816E+02	<b>8.0114E-49</b>
	平均值	1.8092E+07	4.3598E+00	3.8097E+00	6.7896E+00	<b>4.0070E-50</b>
	标准差	5.5148E+07	1.2895E+00	6.4042E-01	2.6217E+01	<b>1.7914E-49</b>
<b>F3</b>	最优值	2.7705E+04	2.5733E+00	3.5695E-01	1.1125E+02	<b>5.7613E-138</b>
	最差值	8.8552E+04	7.8447E+00	9.2086E-01	3.0820E+02	<b>1.1954E-79</b>
	平均值	5.1820E+04	5.0493E+00	5.9536E-01	2.0522E+02	<b>6.0108E-81</b>
	标准差	1.6682E+04	1.4706E+00	1.4290E-01	6.2326E+01	<b>2.6722E-80</b>
<b>F4</b>	最优值	6.3727E+01	7.3870E-01	3.3337E-01	1.2551E+00	<b>5.8467E-75</b>
	最差值	7.7879E+01	1.2198E+00	5.8009E-01	3.5464E+00	<b>4.0606E-41</b>
	平均值	7.0004E+01	8.7301E-01	4.5841E-01	2.0899E+00	<b>2.1190E-42</b>
	标准差	3.8411E+00	1.0763E-01	6.2636E-02	6.3651E-01	<b>9.0633E-42</b>
<b>F5</b>	最优值	3.4712E+04	4.4565E+01	8.4212E+01	3.3752E+01	<b>1.4957E-07</b>
	最差值	5.4196E+06	1.5287E+02	1.6467E+02	2.6561E+03	<b>7.1850E+00</b>
	平均值	1.4936E+06	8.6509E+01	1.2555E+02	4.9915E+02	<b>2.5181E+00</b>
	标准差	1.2986E+06	3.2110E+01	2.4406E+01	7.8531E+02	<b>2.1465E+00</b>
<b>F6</b>	最优值	2.5284E+04	1.2648E-01	1.2498E+00	8.0899E-01	<b>1.0896E-10</b>
	最差值	5.1247E+04	3.2947E-01	3.1529E+00	1.9608E+00	<b>2.7168E-10</b>
	平均值	3.9766E+04	2.0909E-01	1.9834E+00	1.2461E+00	<b>1.6009E-10</b>
	标准差	7.0489E+03	6.4236E-02	4.9561E-01	3.6952E-01	<b>3.9965E-11</b>

**Table 3.** Results of multi-modal benchmark functions  
**表 3.** 多峰函数实验结果

函数	结果	BA	CS	FPA	MVO	LSMVO
F7	最优值	1.2063E+02	4.3862E+01	1.5553E+00	7.0478E+01	0
	最差值	2.6397E+02	6.5374E+01	6.2159E+01	1.6060E+02	0
	平均值	2.0633E+02	5.3863E+01	2.5231E+01	1.1398E+02	0
	标准差	4.0418E+01	6.4591E+00	1.6720E+01	2.3051E+01	0
F8	最优值	1.8824E+01	2.5968E+00	1.2640E+00	5.3806E-01	8.8818E-16
	最差值	1.9953E+01	3.5927E+00	2.2865E+00	3.0380E+00	8.8818E-16
	平均值	1.9211E+01	3.0461E+00	1.7459E+00	1.6963E+00	8.8818E-16
	标准差	3.3019E-01	3.2588E-01	2.4307E-01	5.9864E-01	0
F9	最优值	4.7801E+02	1.0271E-02	1.2722E-02	7.1059E-01	0
	最差值	6.6798E+02	3.6964E-02	4.8351E-02	1.0100E+00	0
	平均值	5.7293E+02	2.0333E-02	2.8755E-02	8.7723E-01	0
	标准差	4.8753E+01	6.9765E-03	9.5901E-03	7.4760E-02	0
F10	最优值	1.5742E+06	1.3387E-02	5.6834E-02	9.9508E-02	2.2006E-13
	最差值	1.8804E+07	1.6785E-01	1.6051E-01	3.5621E+00	5.2129E-12
	平均值	8.2380E+06	6.3029E-02	1.0838E-01	1.6166E+00	2.7525E-12
	标准差	5.9157E+06	4.2813E-02	3.4225E-02	9.7694E-01	1.3039E-12
F11	最优值	5.4082E+06	3.1026E+00	5.7330E-01	7.4449E-02	2.4393E-12
	最差值	9.0013E+07	5.7181E+00	1.8728E+00	4.8664E-01	1.1289E+00
	平均值	5.0547E+07	4.4302E+00	1.2610E+00	2.1568E-01	7.9103E-02
	标准差	2.2612E+07	7.3259E-01	3.5735E-01	1.2039E-01	2.6701E-01

由表 2 可见, 在单峰函数方面: 对于函数  $F_1$ , LSMVO 的最优值精度达到了  $e^{-147}$ , 相较于相对优秀的 CS 高出 146 个数量级; 在平均值方面, LSMVO 的求解精度为  $e^{-75}$ , 比仅次于它的 CS 高出 74 个数量级。对于函数  $F_2$ , LSMVO 的最优值精度达到了  $e^{-85}$ , 较排名第二的 MVO 提升 84 个数量级; 在平均值方面, LSMVO 的求解精度均为  $e^{-50}$ , 比 MVO 提高了 50 个数量级。对于函数  $F_3$ , LSMVO 的最优值和平均值精度达到了  $e^{-138}$  和  $e^{-81}$ , 比 FPA 的最优值和平均值高 137 个和 80 个数量级。对于函数  $F_4$ , LSMVO 的最优值和平均值精度达到了  $e^{-75}$  和  $e^{-42}$ , 比 FPA 和 CS 的最优值和平均值高 74 个和 41 个数量级。对于函数  $F_5$ , LSMVO 的最优值和平均值精度达到了  $e^{-7}$  和  $e^0$ , 比仅次于它的 CS 的最优值和平均值高 8 个和 1 个数量级。对于函数  $F_6$ , LSMVO 的最优值和平均值精度均达到了  $e^{-10}$ , 比仅次于它的 CS 均高出 9 个数量级。

多峰函数优化方面: 对于函数  $F_7$ , LSMVO 的最优值、最差值、平均值、标准差均为 0, 性能显著优于 BA、CS、FPA 和 MVO。对于函数  $F_8$ , LSMVO 的最优值、最差值和平均值精度达到了  $e^{-16}$ , 明显优于其他对比算法, 且其标准差为 0, 说明其具有很强的稳定性。对于函数  $F_9$ , LSMVO 的最优值、最差值、平均值、标准差均为 0, 这是 BA、CS、FPA 和 MVO 所无法达到的。对于函数  $F_{10}$ , LSMVO 的最优值和平均值精度达到了  $e^{-13}$  和  $e^{-12}$ , 较 CS 和 FPA 的对应指标高出 11 个和 10 个数量级。对于函数  $F_{11}$ ,

LSMVO 的最优值和平均值精度达到了  $e^{-12}$  和  $e^{-2}$ ，较仅次于它的 MVO 分别高出 10 个和 1 个数量级；不过，LSMVO 在最差值和标准差指标上略逊于 MVO。

综合来看，无论是单峰函数还是在多峰函数优化问题，LSMVO 的性能均优于 BA、CS、FPA 和 MVO。

图 3~13 为 BA、CS、FPA 和 MVO 及 LSMVO 在求解函数  $F_1 \sim F_{11}$  最优值时函数随迭代次数变化的曲线图。

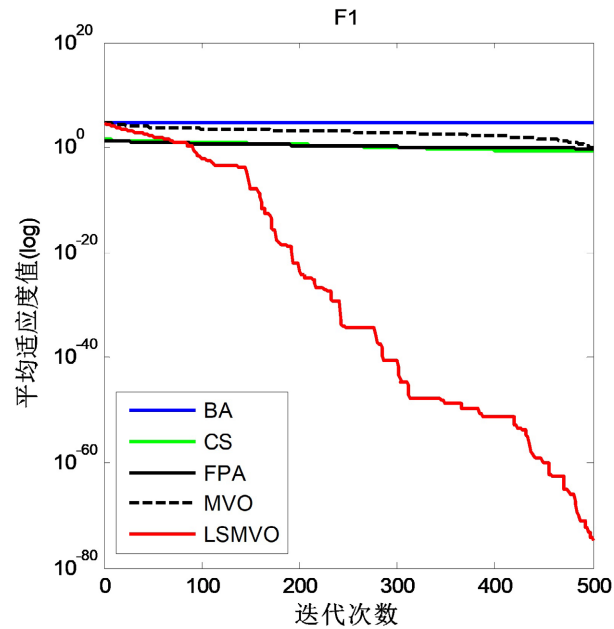


Figure 3. Evolution curve of fitness for F1

图 3. F1 的收敛曲线对比

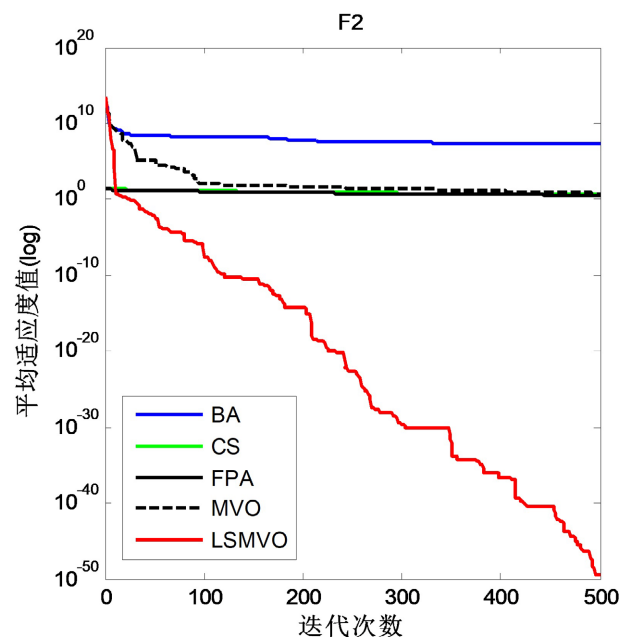


Figure 4. Evolution curve of fitness for F4

图 4. F2 的适应度函数收敛曲线



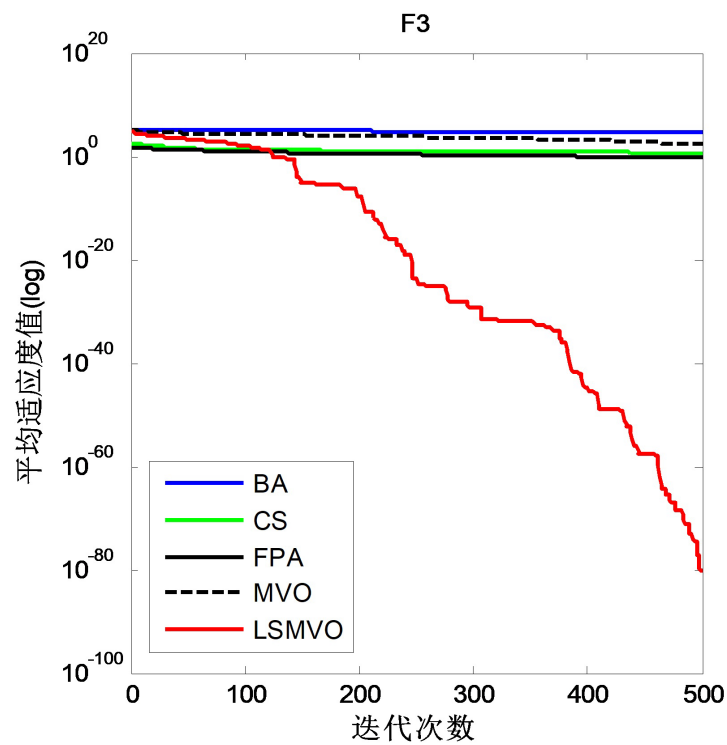


Figure 5. Evolution curve of fitness for F3

图 5. F3 的适应度函数收敛曲线

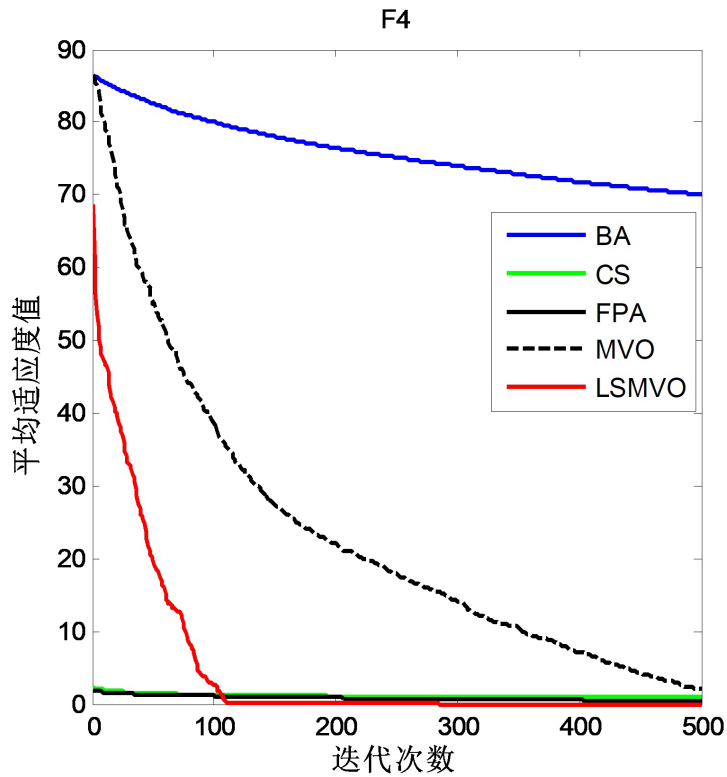


Figure 6. Evolution curve of fitness for F4

图 6. F4 的使用度函数收敛曲线

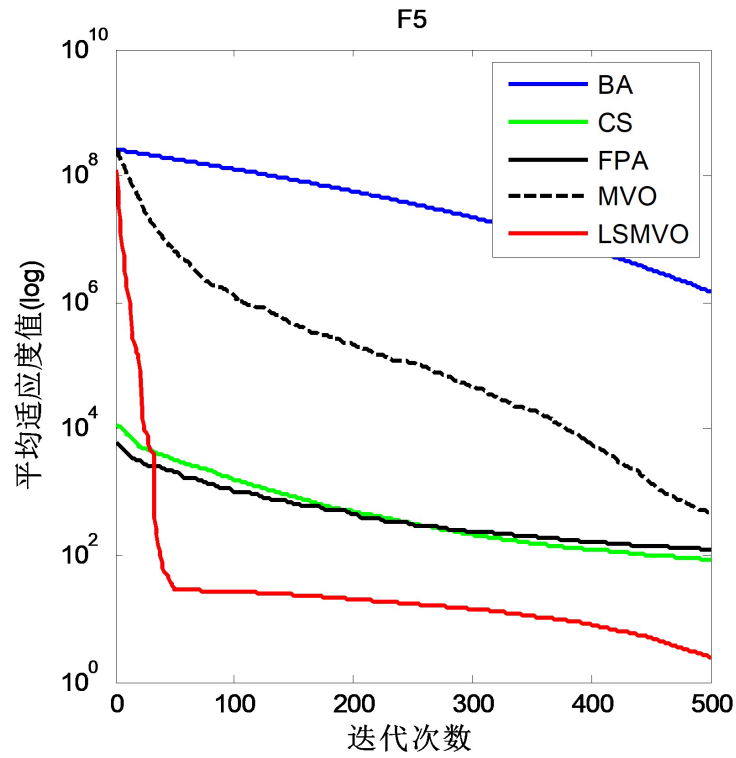


Figure 7. Evolution curve of fitness for F5

图 7. F5 的适应度函数收敛曲线

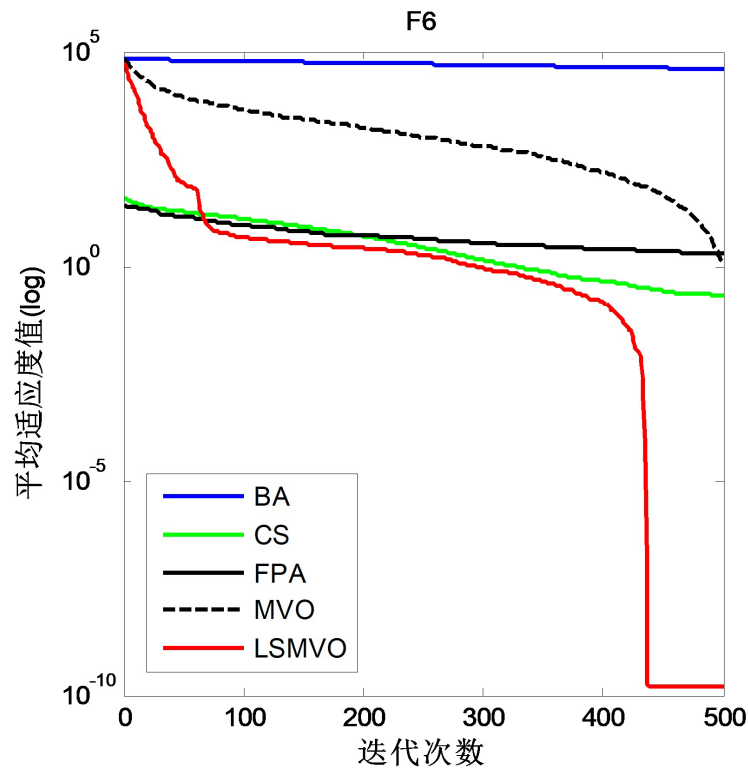


Figure 8. Evolution curve of fitness for F6

图 8. F6 的适应度函数收敛曲线

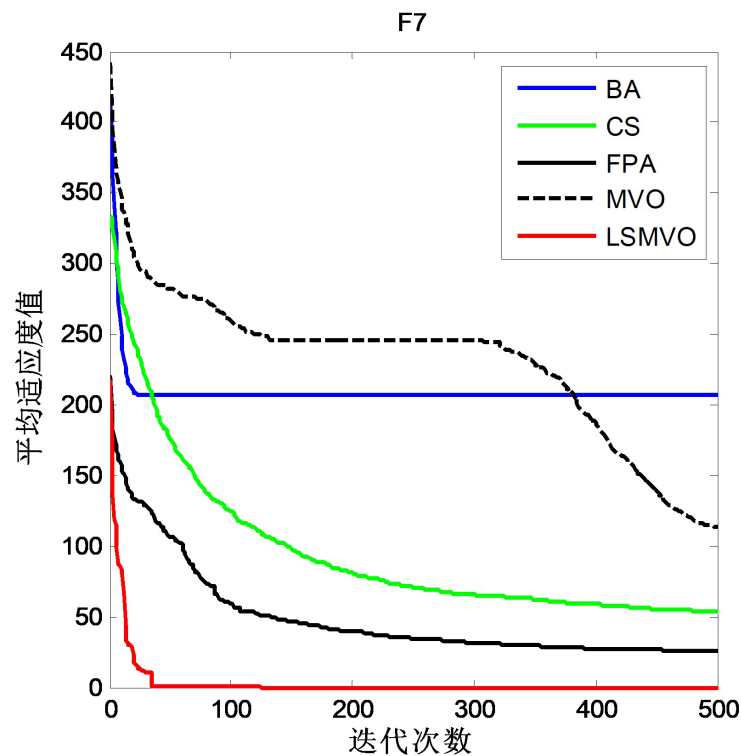


Figure 9. Evolution curve of fitness for F7

图 9. F7 的适应度函数收敛曲线

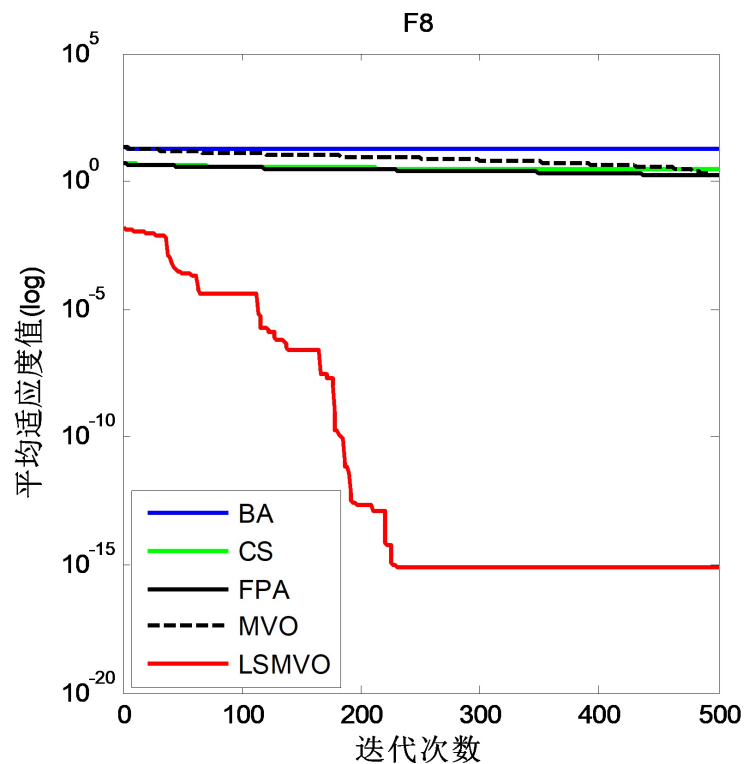


Figure 10. Evolution curve of fitness for F8

图 10. F8 的适应度函数收敛曲线

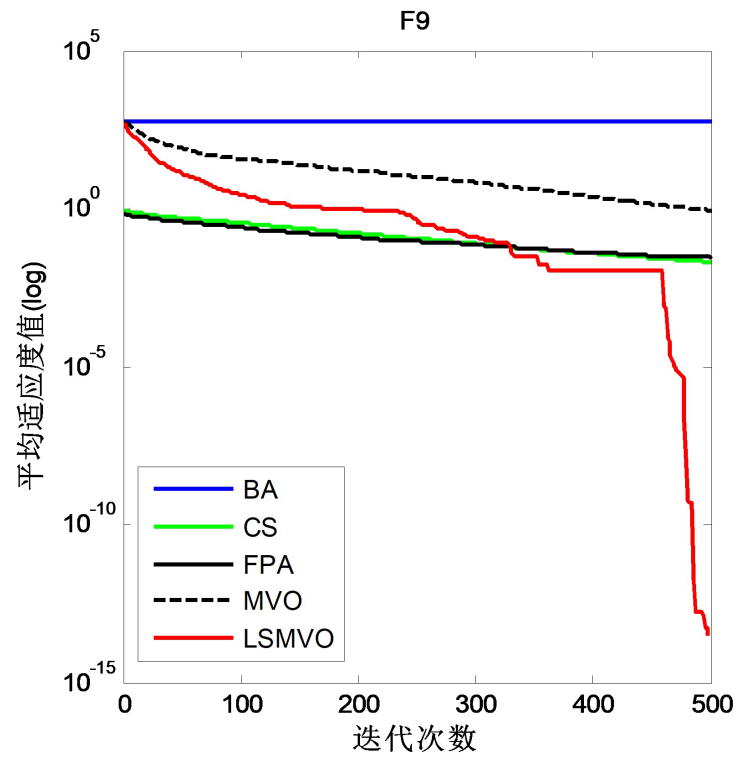


Figure 11. Evolution curve of fitness for F9

图 11. F9 的适应度函数收敛曲线

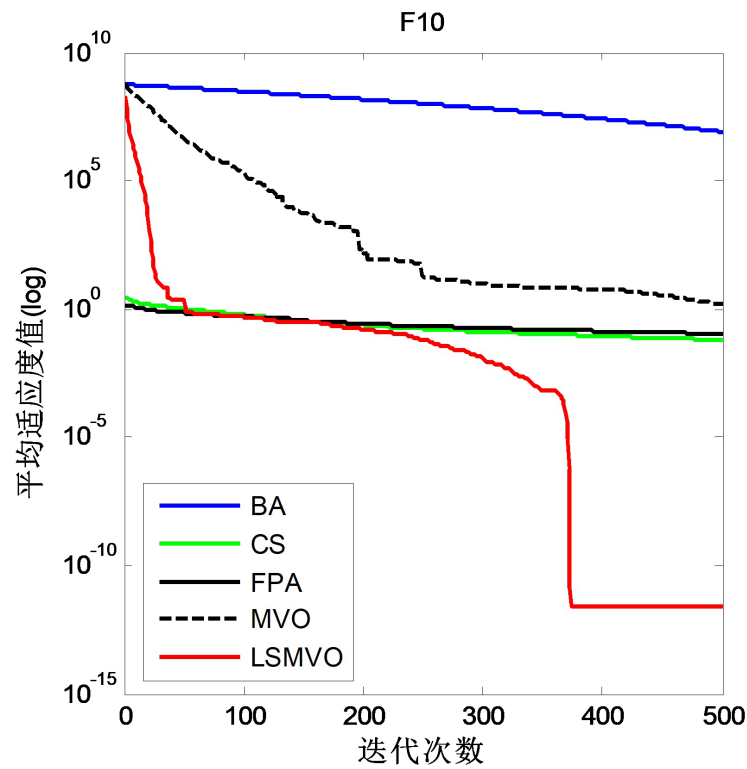


Figure 12. Evolution curve of fitness for F10

图 12. F10 的适应度函数收敛曲线

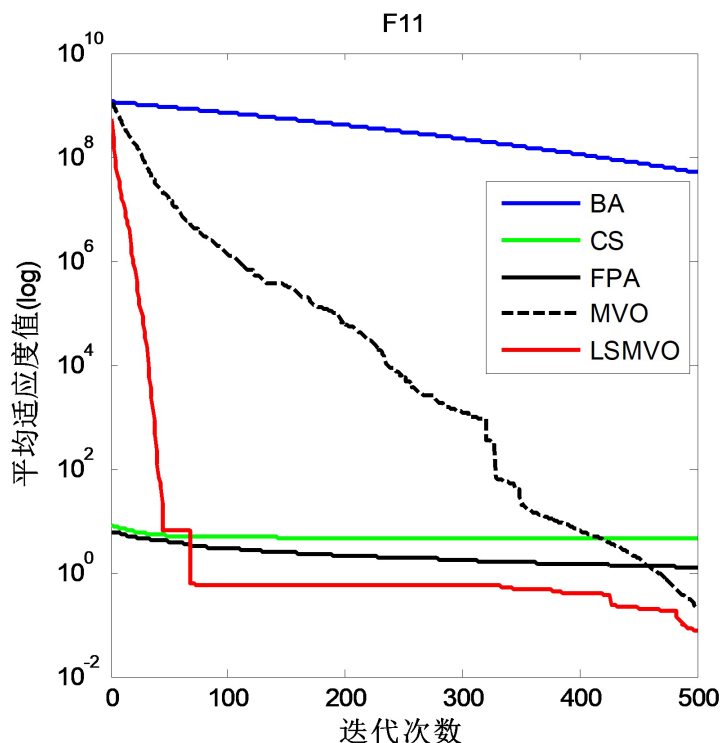


Figure 13. Evolution curve of fitness for F11

图 13. F11 的适应度函数收敛曲线

从图 3~13 可以进一步验证 LSMVO 的优越性。从图 3、图 4、图 5、图 10 可以看出，当其他算法已接近收敛状态时，LSMVO 仍保持继续寻优的能力。上述函数优化结果表明，加入单纯形法和 Lévy 飞行策略的 LSMVO 在函数优化中展现出较强优势，其测试结果更接近理论最优值，证实了该改进算法在函数优化问题中的可行性与有效性。

#### 4. LSMVO 解决焊接梁设计问题

现实中的工程优化问题大多是非线性的，而且需要满足一些复杂的，甚至是刁钻的约束条件。这些约束条件大都是由一些非线性方程组组成的，这使得解的空间变得越发复杂，包括一些经典的算法都很容易陷入到局部最优解中。解决约束优化问题的主要任务是处理约束条件，能找到问题的可行解是非常不容易的。为了处理这些带有约束的工程优化问题，所使用的方法多种多样，最常用的就是罚函数法，它的目标是构造罚函数，将有约束条件的问题转化为求解无约束的最优化问题。

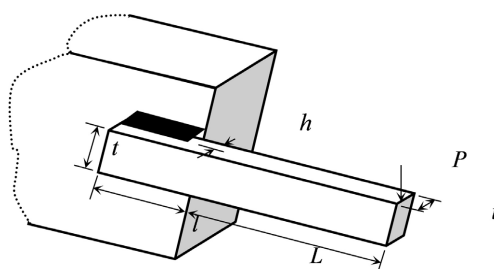


Figure 14. Welded beam design

图 14. 焊接梁设计模型图

焊接梁设计是需要将矩形梁焊接成悬臂梁以承受一定的载荷，设计焊接梁的目的是使制造的总成本最小化。焊接梁的约束条件包括剪应力  $\tau$ ，压曲临界荷载  $P_c$ ，终端挠度  $\delta$  和法向应力  $\sigma$ 。其中，最大焊接剪应力  $\tau_{\max} = 13600$  psi，最大法向应力  $\sigma_{\max} = 30000$  psi。这里使用 LSMVO 解决了焊接梁设计问题，以进一步验证新算法的性能。焊接梁设计模型如图 14 所示。

焊接梁设计问题有四个变量，分别是焊缝厚度  $h$ ，棒的附接部分的长度  $l$ ，棒的高度  $t$  和棒的厚度  $b$ 。焊接梁设计问题的公式描述如下：

$$\bar{x} = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b]$$

最小化

$$f(\bar{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

约束条件

$$g_1(\bar{x}) = \tau(\bar{x}) - \tau_{\max} \leq 0,$$

$$g_2(\bar{x}) = \sigma(\bar{x}) - \sigma_{\max} \leq 0,$$

$$g_3(\bar{x}) = \delta(\bar{x}) - \delta_{\max} \leq 0,$$

$$g_4(\bar{x}) = x_1 - x_4 \leq 0,$$

$$g_5(\bar{x}) = P - P_c(\bar{x}) \leq 0,$$

$$g_6(\bar{x}) = 0.125 - x_1 \leq 0,$$

$$g_7(\bar{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0,$$

变量取值范围

$$0.1 \leq x_1 \leq 2,$$

$$0.1 \leq x_2 \leq 10,$$

$$0.1 \leq x_3 \leq 10,$$

$$0.1 \leq x_4 \leq 2,$$

此处

$$\tau(\bar{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right),$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},$$

$$\sigma(\bar{x}) = \frac{6PL}{x_4x_3^2}, \delta(\bar{x}) = \frac{6PL^3}{Ex_3^2x_4}$$

$$P_c(\bar{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$
$$P = 6000lb,$$
$$L = 14in.,$$
$$\delta_{\max} = 0.25in.,$$
$$E = 30 \times 10^6 psi,$$
$$G = 12 \times 10^6 psi,$$
$$\tau_{\max} = 13600 psi,$$
$$\sigma_{\max} = 30000 psi,$$

针对焊接梁设计问题, Coello [11]和 Deb [12] [13]使用 GA 解决此问题, 而 Lee 和 Geem [14]使用 HS 解决了这个问题。Ragsdell 和 Philips 采用了 Richardson 的随机方法, 单纯形法, Davidon-Fletcher-Powell, Griffith 和 Stewart 的连续线性近似的数学方法应用到解决焊接梁设计问题[15]。表 4 给出了焊接梁设计问题的比较结果。从该表可以看出, 与其他算法相比 LSMVO 能够取得比较好的设计方案。LSMVO 得到的最优值为:

$$f(\bar{x}) = 1.18122$$

对应的最优解为:

$$\bar{x} = [0.2111, 3.3454, 9.3483, 0.2112]$$

**Table 4.** Comparison results of the welded beam design problem  
**表 4.** 焊接梁设计问题结果比较

Algorithm	Optimum variables				Optimum cost
	<i>H</i>	<i>l</i>	<i>t</i>	<i>B</i>	
<b>LSMVO</b>	<b>0.2111</b>	<b>3.3454</b>	<b>9.3483</b>	<b>0.2112</b>	<b>1.8122</b>
GA (Coello)	N/A	N/A	N/A	N/A	1.8245
GA (Deb)	N/A	N/A	N/A	N/A	2.3800
GA (Deb)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem)	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815

5. 结束语

针对多元宇宙优化算法在迭代后期探索能力不足的问题, 本文构造一种基于单纯性和 Lévy 飞行策略的多元宇宙优化算法。实验测试结果表明, 与原多元宇宙优化算法相比, 新算法的寻优能力得到了较大提升。将新算法应用于焊接梁设计中时, 也取得了更好的实验效果。对于未来的工作研究, 一方面将进

一步拓展多元宇宙优化算法在更多实际工程领域的应用场景；另一方面将致力于开发更有效的新型元启发式优化方法，以解决复杂优化问题中的技术难题。

## 基金项目

广西重点研发计划项目(桂科 AB25069262)；广西高校中青年教师科研基础能力提升项目(2024KY0871)；广西科技师范学院科研项目(GXKS2024YB033, GXKS2024YB032, GXKS2025YB032, GXKS2025YB042, GXKSKYPT2025005, GXKSKYPT2025008)。

## 参考文献

- [1] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 1997.
- [2] Alyasseri, Z.A.A., Alomari, O.A., Al-Betar, M.A., Makhadmeh, S.N., Doush, I.A., Awadallah, M.A., *et al.* (2022) Recent Advances of Bat-Inspired Algorithm, Its Versions and Applications. *Neural Computing and Applications*, **34**, 16387-16422. <https://doi.org/10.1007/s00521-022-07662-y>
- [3] Huang, S., Lin, N., Wang, Z., Zhang, Z., Wen, S., Zhao, Y., *et al.* (2024) A Novel Data-Driven Method for Online Parameter Identification of an Electrochemical Model Based on Cuckoo Search and Particle Swarm Optimization Algorithm. *Journal of Power Sources*, **601**, Article ID: 234261. <https://doi.org/10.1016/j.jpowsour.2024.234261>
- [4] Mergos, P.E. and Yang, X. (2022) Flower Pollination Algorithm with Pollinator Attraction. *Evolutionary Intelligence*, **16**, 873-889. <https://doi.org/10.1007/s12065-022-00700-7>
- [5] Mirjalili, S., Mirjalili, S.M. and Hatamlou, A. (2015) Multi-Verse Optimizer: A Nature-Inspired Algorithm for Global Optimization. *Neural Computing and Applications*, **27**, 495-513. <https://doi.org/10.1007/s00521-015-1870-7>
- [6] Abasi, A., Khader, A.T. and Al-Betar, M.A. (2022) An Improved Multi-Verse Optimizer for Text Documents Clustering. *Kufa Journal of Engineering*, **13**, 28-42. <https://doi.org/10.30572/2018/kje/130203>
- [7] Kumar, S., Panagant, N., Tejani, G.G., Pholdee, N., Bureerat, S., Mashru, N., *et al.* (2023) A Two-Archive Multi-Objective Multi-Verse Optimizer for Truss Design. *Knowledge-Based Systems*, **270**, Article ID: 110529. <https://doi.org/10.1016/j.knsys.2023.110529>
- [8] Son, P.V.H. and Nguyen Dang, N.T. (2023) Solving Large-Scale Discrete Time-cost Trade-Off Problem Using Hybrid Multi-Verse Optimizer Model. *Scientific Reports*, **13**, Article No. 1987. <https://doi.org/10.1038/s41598-023-29050-9>
- [9] 王玉芳, 程培浩, 闫明. 融合信赖域与非线性单纯形法的黑翅鸢优化算法[J]. 计算机科学与探索, 2025, 19(7): 1789-1807.
- [10] Iacca, G., dos Santos Junior, V.C. and Veloso de Melo, V. (2021) An Improved Jaya Optimization Algorithm with Lévy Flight. *Expert Systems with Applications*, **165**, Article ID: 113902. <https://doi.org/10.1016/j.eswa.2020.113902>
- [11] Coello Coello, C.A. (2000) Constraint-Handling Using an Evolutionary Multiobjective Optimization Technique. *Civil Engineering and Environmental Systems*, **17**, 319-346. <https://doi.org/10.1080/02630250008970288>
- [12] Deb, K. (1991) Optimal Design of a Welded Beam via Genetic Algorithms. *AIAA Journal*, **29**, 2013-2015. <https://doi.org/10.2514/3.10834>
- [13] Deb, K. (2000) An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, **186**, 311-338. [https://doi.org/10.1016/s0045-7825\(99\)00389-8](https://doi.org/10.1016/s0045-7825(99)00389-8)
- [14] Lee, K.S. and Geem, Z.W. (2005) A New Meta-Heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice. *Computer Methods in Applied Mechanics and Engineering*, **194**, 3902-3933. <https://doi.org/10.1016/j.cma.2004.09.007>
- [15] Ragsdell, K.M. and Phillips, D.T. (1976) Optimal Design of a Class of Welded Structures Using Geometric Programming. *Journal of Engineering for Industry*, **98**, 1021-1025. <https://doi.org/10.1115/1.3438995>