

文章编号: 2095-2163(2024)01-0016-06

中图分类号: TP301.6

文献标志码: A

# 基于精英思想自适应改进万有引力搜索算法

刘诗琪, 潘大志

(西华师范大学 数学与信息学院, 四川 南充 637009)

**摘要:** 为了解决万有引力搜索算法容易出现局部最优的问题,提出了一种新型改进万有引力搜索算法。该算法在质量的计算中引入随机因子,结合精英思想,基于适应度值对力进行有选择的合成,并且对更优粒子对应的力赋予更大的随机数;引入控制参数,自适应地更新粒子的位置,减小某些粒子过于随意变化带来的影响。通过以上这些操作,增强了算法的随机性,同时保证了算法的收敛性。经对10个基准函数进行仿真实验,结果表明新算法有更好的收敛速度和寻优精度,全局和局部优化能力增强。

**关键词:** 万有引力搜索算法;精英思想;自适应;随机因子;函数优化

## Adaptive improvement of gravitational search algorithm based on elite thought

LIU Shiqi, PAN Dazhi

(School of Mathematics &amp; Information, China West Normal University, Nanchong Sichuan 637009, China)

**Abstract:** In order to solve the problem that the gravitational search algorithm is prone to local optimization, a new improved gravitational search algorithm is proposed. The algorithm introduces a random factor in the calculation of mass; combines the idea of elite, selectively synthesizes the force based on the value of fitness, assigns a larger random number to the force corresponding to the better particles; and introduces a control parameter to adaptively update the position of the particles to reduce the impact of the overly random changes of some particles. Through these operations, the randomness of the algorithm is enhanced and the convergence of the algorithm is guaranteed. After simulation experiments on 10 benchmark functions, the results show that the new algorithm has better convergence speed and optimization accuracy, and the global and local optimization ability is enhanced.

**Key words:** gravitational search algorithm; elite thought; self-adaption; random factor; function optimization

## 0 引言

启发式优化算法是优化理论的重要组成部分,研究人员从自然环境中获得灵感,将生物习性融入模型之中,提出了许多启发式优化算法,如:蚁群算法(Ant Colony Optimization, ACO)、遗传算法(Genetic Algorithm, GA)等。万有引力搜索算法(Gravitational Search Algorithm, GSA)是由Esmat Rashedi等人<sup>[1]</sup>于2009年根据物理学中的万有引力定律和牛顿第二定律提出的一种新型启发式智能优化算法,该算法具有全局搜索能力强,收敛速度快等优点,相比于遗传算法、粒子群算法(Particle Swarm Optimization, PSO)等其整体性能具有较好的竞争力。GSA算法已经成功应用于诸多领域,例如工程

设计优化、图像识别、电力系统负荷控制与预测等。Rather Sajad Ahmad等<sup>[2-3]</sup>在GSA算法中分别融入Lévy飞行和混沌搜索,对减速器设计、多盘离合器制动器设计等工程中的变量最小化问题进行求解;Thiagarajan Kowsalya等<sup>[4]</sup>利用分层集成学习和基于相关系数的引力搜索算法进行卫星图像分类;Kumar Yogesh等<sup>[5]</sup>在融合了量子计算的基础上,进一步加入深度神经网络对图像进行识别分类;Gonggui Chen等<sup>[6]</sup>利用GSA改进模糊控制器的参数,优化电力控制系统;Khanesar Mojtaba Ahmadi等<sup>[7]</sup>将GSA与最小二乘算法训练的直觉模糊逻辑系统结合,用于电力负荷的短期预测。还有一些学者在GSA算法的基础上融合其他算法或思想,提出了一些新算法,如Amponsah Alfred Adutwum等<sup>[8]</sup>提

**基金项目:** 国家自然科学基金(11871059);四川省教育厅自然科学基金(18ZA0469);西华师范大学英才科研基金(17YC385);西华师范大学2022年校级大学生创新创业训练计划项目(cxcy2022030)。

**作者简介:** 刘诗琪(1999-),女,硕士研究生,主要研究方向:智能计算。

**通讯作者:** 潘大志(1974-),男,博士,教授,主要研究方向:智能计算、算法设计。Email:pdzjz@126.com

**收稿日期:** 2023-01-09

出了基于引力搜索算法的多领导综合学习粒子群算法;Wang Yirui 等<sup>[9]</sup>提出了具有层次结构和分布式框架的 GSA 算法;傅文渊<sup>[10]</sup>提出了具有万有引力加速机制的布谷鸟算法。

但是 GSA 也存在易出现早熟现象和局部最优等问题。针对于此,张维平等<sup>[11]</sup>通过反向学习策略初始化种群、精英策略和边界变异策略增强了全局搜索能力;马力等<sup>[12]</sup>和刘小刚等<sup>[13]</sup>对融合了 PSO 算法的 GSA 算法进行进一步改进,分别是动态调整搜索边界和改进惯性权重及学习因子。

本文利用 GSA 算法的搜索特征,提出一种新型改进 GSA 的算法。为了验证算法的有效性,选择了 10 个常见的基本测试函数进行实验,并将其与基本的 GSA 算法、PSO 算法和文献[13]中的 IPSOGSA 算法进行了对比分析。

## 1 基本万有引力搜索算法

### 1.1 基本原理

GSA 的基本原理是粒子间的相互作用。由万有引力定律可知,任意的两个粒子间都存在引力作用;再由牛顿第二定律可知,质量越大的物体运动得越慢,因此质量小的粒子会朝着质量大的粒子方向运动。用粒子的位置表示优化问题的解,随着算法的迭代,在引力的作用下,粒子在空间中不断地搜索,粒子的位置不断发生改变,最终整个群体都聚集在质量最大的粒子周围。当算法运行结束后,质量最大的粒子的位置就代表优化问题的最优解。

### 1.2 数学模型

假设在  $n$  维搜索空间中有  $N$  个粒子,定义  $t$  时刻第  $i$  个粒子的位置如式(1):

$$X_i(t) = (x_i^1(t), x_i^2(t), \dots, x_i^k(t), \dots, x_i^n(t)) \quad (1)$$

其中,  $i = 1, 2, \dots, N$ ,  $x_i^k(t)$  表示  $t$  时刻粒子  $i$  在第  $k$  维空间中的位置。

$M_i(t)$  表示粒子  $i$  在  $t$  时刻的质量,计算如式(3):

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (2)$$

$$M_i(t) = m_i(t) / \sum_{j=1}^N m_j(t) \quad (3)$$

其中,  $fit_i(t)$  表示粒子  $i$  在  $t$  时刻的适应度值,  $best(t)$  和  $worst(t)$  分别表示在  $t$  时刻粒子群体中的最优适应度值和最差适应度值。

在  $t$  时刻,粒子  $i$  在第  $k$  维空间上受到粒子  $j$  的引力大小如式(4):

$$F_{ij}^k(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^k(t) - x_i^k(t)) \quad (4)$$

其中,  $R_{ij}(t)$  表示粒子  $i$  和粒子  $j$  之间的欧式距离;  $G(t)$  表示  $t$  时刻的引力常数;  $\varepsilon$  为一个很小的正数。计算如式(5):

$$G(t) = G_0 e^{-\alpha/T} \quad (5)$$

其中,  $G_0$  为初始时粒子的引力系数,  $\alpha$  为控制参数。

定义粒子  $i$  在第  $k$  维空间上受到的合力如式(6):

$$F_i^k(t) = \sum_{j=1, j \neq i}^N r_j F_{ij}^k(t) \quad (6)$$

其中,  $r_j$  为  $(0, 1)$  的随机数。

在  $t$  时刻,粒子  $i$  在第  $k$  维空间上的加速度如式(7);粒子  $i$  在第  $k$  维空间上的速度和位置更新公式分别为式(8)、式(9)。

$$a_i^k(t) = \frac{F_i^k(t)}{M_i(t)} \quad (7)$$

$$v_i^k(t+1) = r_i \times v_i^k(t) + a_i^k(t) \quad (8)$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \quad (9)$$

## 2 改进的万有引力搜索算法

### 2.1 带有随机因子的质量公式

在 GSA 算法中,质量由适应度值决定,与当前种群的最优值和最差值相关,直接反映出适应度值的分布情况,当  $m_i(t)$  的值越接近于 1 时,当前个体越靠近本次迭代的最优值。由于 GSA 算法容易出现早熟收敛,因此对质量增加随机权重,增加算法的随机性,在一定程度上跳出局部最优。改进后的质量计算如式(11):

$$\tilde{m}_i(t) = r_i m_i(t) \quad (10)$$

$$\tilde{M}_i(t) = \frac{\tilde{m}_i(t)}{\sum_{j=1}^N \tilde{m}_j(t)} \quad (11)$$

其中,  $r_i$  ( $r_i \in (0, 1)$ ) 为随机数。

### 2.2 基于精英思想自适应选择合力

由万有引力定律可知,每个粒子都受到其他所有粒子的吸引,因此当大多数粒子处于某个较优位置时,剩下的粒子也难以跳出这个范围,整个种群容易陷入局部最优。为解决这一问题,本文利用精英的思想,结合适应度值排序,自适应地选择部分力进行合成。下面以最小化问题为例说明该算子。

首先,对最小化问题中的目标函数值进行升序排序,得到对应的粒子序号  $b = (b_1, b_2, \dots, b_N)$ ; 生

成一组(0,1)的随机数并降序排序得到  $p = (p_1, p_2, \dots, p_N)$ 。按照目标函数值由小到大的顺序,给较优的粒子赋予更大的随机数。

文献[1]中设置参数  $K$  来控制合成力的数量。利用迭代次数使  $K$  值从  $N$  线性递减到 1,随着算法迭代的进行,合成力的数量逐渐减少,粒子  $i$  受到的牵引作用减少,粒子飞行的随机性增强,范围扩大,这种方式能够有效地跳出局部最优。

但是由于算法后期合成力的粒子太少,难以将大部分粒子吸引到最优位置附近,所以粒子的位置会比较分散,解的稳定性难以保证。因此本文重新设置参数  $K$ , 如式(12):

$$K = \max \{ \text{floor}(N \cdot (\cos(\frac{\pi}{2} \cdot \frac{t+T}{T}) + 1) + 1), m \} \quad (12)$$

改进后的合力公式如式(13):

$$\tilde{F}_i^k(t) = \sum_{j=1, j \neq i}^K p_j F_{ib_j}^k(t) \quad (13)$$

其中,  $\text{floor}()$  为向下取整函数,  $m$  为设定的一个常数。

改进后的  $K$  从  $N$  非线性递减到  $m$  后保持不变。参数  $K$  的前半部分取值是由凹函数变化得到的,即减小的越来越慢。在算法迭代初期,选择较多的力进行合成,在合力的引导下,算法能够快速收敛到较优值附近;随着迭代的进行,合成力的粒子数逐渐减少,因此粒子  $i$  受到其他粒子的影响也随之减少,粒子移动的范围扩大,找到更优解的可能性增加,能够有效地避免算法陷入局部最优。在算法的后期,只选择最优的前  $m$  个粒子对应的力进行合成,既增加了算法的随机性,同时也保证了算法的收敛性和解的稳定性。在本文中,通过实验得知  $m$  取值为 20% $N$  时综合效果最好。

### 2.3 自适应位置更新公式

由于粒子位置表示问题的解,因此位置更新公式直接决定解的情况。在基本 GSA 算法中,粒子下一次的位置直接由上一次的位置与粒子速度之和来确定。当粒子移动步长较大时,可能会远离最优解;当粒子移动步长较小时,可能难以搜索到更好的解,容易陷入局部最优。

为解决这一问题,文献[14]引入收缩因子  $\mu$  和  $c$ , 计算如式(14)和式(15),  $\mu$  和  $c$  均随着迭代次数的增加而减小,取值范围均为(0,1)。在迭代初期,粒子的移动步长较大,利用收缩因子能够快速收敛到较优粒子周围;在迭代后期,粒子的移动步长减小,收缩因

子帮助粒子在最优值附近进行深度搜索。

$$\mu = e^{-n \times (t/T)^w} \quad (14)$$

$$c = 1 - \frac{t}{T + \text{betarnd}} \quad (15)$$

其中,  $w$  在文献[14]中取值为 10,  $\text{betarnd}$  是  $B$  分布产生的随机数。

通过计算可以得知,当算法的迭代次数达到最大值时,收缩因子  $\mu$  和  $c$  的值都非常小且趋于 0,这会在很大程度上影响解的精度,不能说明解是由于算法搜索得到的还是因为收缩因子很小影响得到的。因此,对收缩因子进行改进,  $\mu$  的改进如式(16)。通过计算可知,改进后的  $\mu$  取值范围是  $(e^{-2}, 1)$ ,  $c$  借助文献[15]中设置的惯性权重,重新进行设置,  $c$  的表达式如式(17):

$$\mu = e^{-\frac{2t}{T}} \quad (16)$$

$$c = 2 \cdot \left( -\frac{c_{\max} - c_{\min}}{T} \cdot t \right) + \frac{c_{\max} - c_{\min}}{1 + e^{-\left(\frac{32}{T}t - 16\right)}} + c_{\max} \quad (17)$$

改进后的位置更新公式如式(18):

$$x_i^k(t+1) = \mu x_i^k(t) + cv_i^k(t+1) \quad (18)$$

其中,  $c_{\max}$  和  $c_{\min}$  控制  $c$  的取值范围,  $c \in [c_{\min}, c_{\max}]$ 。  $\mu$  随着迭代次数的增加而减小,  $c$  随着迭代次数先减小再增大,最后又减小,变化情况如图 1 所示。在迭代初期,粒子仍然保持较大的步长移动,快速移动到较优粒子附近;在迭代中期,由于  $c$  的增加粒子的移动步长适当的增加,向较远的其它地方搜索,找到更优解,避免粒子陷入局部最优;在迭代后期,  $\mu$  和  $c$  都减小,但是只减小到 0.1 附近,粒子的移动步长较小,能够进行深度搜索,同时取值没有过小,保证了算法本身的收敛性,减小了参数本身取值带来的影响。

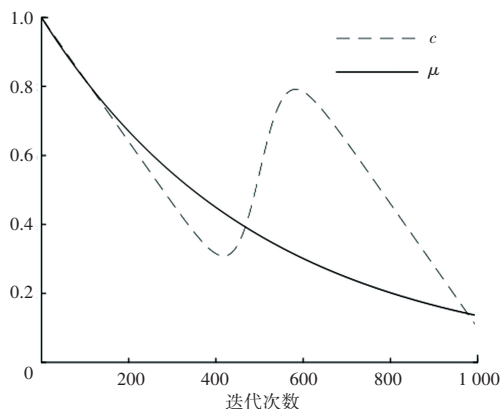


图 1 收缩因子  $\mu$  和  $c$  的变化情况

Fig. 1 Variation curve of contractile factor

### 2.4 IGSA 算法步骤

**Step 1** 算法初始化, 随机生成  $N$  个粒子的速度和位置信息。

**Step 2** 计算每个粒子对应的适应度值; 利用式(2)、式(10)和式(11)计算粒子的质量; 通过式(12)计算参数  $K$ , 再由式(13)计算粒子受到的合力; 由式(7)更新粒子的加速度。

**Step 3** 分别利用式(8)和式(18)更新粒子的速度和位置。

**Step 4** 判断是否达到最大迭代次数: 若是, 执行 Step 5; 若否, 返回 Step 2。

**Step 5** 算法结束, 输出最优值。

### 3 仿真实验

为验证 IGSA 算法的可行性和有效性, 本文选

取常见的 10 个测试函数构成测试集进行测试。其中  $F1 \sim F4$  为单峰测试函数,  $F5 \sim F9$  为多峰测试函数,  $F10$  为固定维度测试函数。10 个测试函数的最优值均为 0。同时将本文算法与 GSA、PSO 及 IPSOGSA 三种算法进行对比分析。

#### 3.1 参数设置

为了实验结果能够有效进行对比, 所有算法均设置相同的种群规模  $N = 50$ 、相同的最大迭代次数  $T = 1\ 000$  和相同的维度  $n = 30$ , 且每种算法都独立运行 30 次。其它参数设置如下:  $m = 10$ ,  $c_{\max} = 1$ ,  $c_{\min} = 0.1$ , 引力常数  $G_0 = 100$ ,  $\alpha = 20$ 。

#### 3.2 测试函数

测试所用函数见表 1。

表 1 测试函数  
Table 1 Test Function

测试函数	定义域
$F1(X) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$
$F2(X) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	$[-10, 10]^n$
$F3(X) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]^n$
$F4(X) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^n$
$F5(X) = \sum_{i=1}^n  x_i \sin(x_i) + 0.1x_i $	$[-10, 10]^n$
$F6(X) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
$F7(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$
$F8(X) = \frac{1}{400}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$
$F9(X) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n \frac{i}{2}x_i\right)^2 + \left(\sum_{i=1}^n \frac{i}{2}x_i\right)^4$	$[-5.14, 5.14]^n$
$F10(X) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2)$	$[-5.14, 5.14]^2$

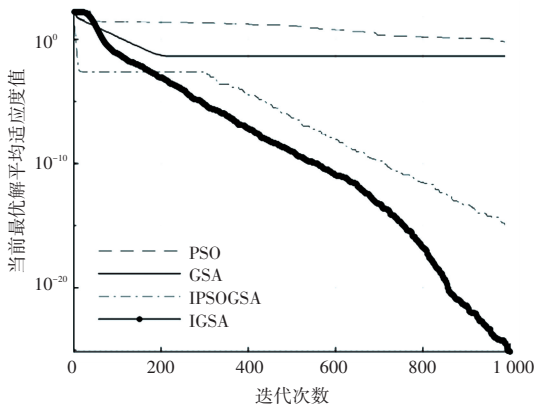
#### 3.3 仿真结果及分析

受篇幅所限, 本文仅选取  $F1$ 、 $F6$ 、 $F8$  和  $F10$  这 4 个函数, 来直观地展示不同函数在 PSO 算法、GSA

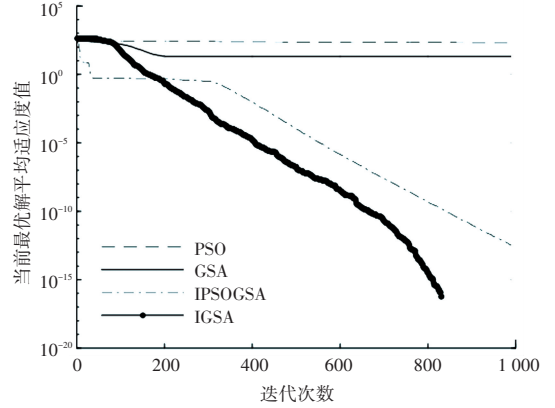
算法、IPSOGSA 算法和 IGSA 算法中的寻优过程, 如图 2 所示。通过函数在不同算法中的收敛曲线可以看出, 无论是求解单峰函数还是多峰函数, 改进后的

IGSA 算法都有更好的寻优精度。虽然在迭代初期算法的收敛速度稍显不足,但是改进后的算法收敛

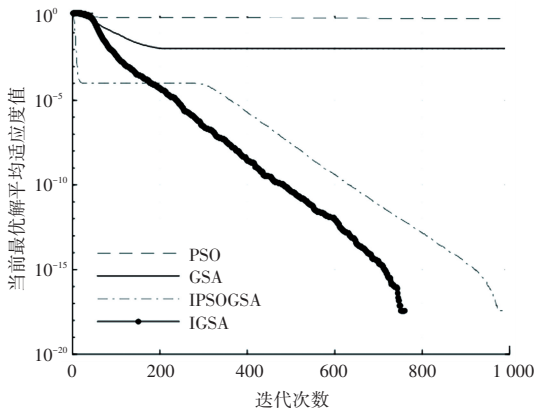
速度更加稳定,算法找到的最优解能够持续性下降,总体来说速度超过 IPSOGSA 算法速度。



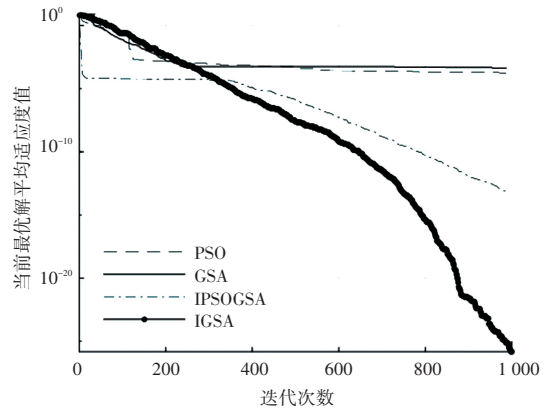
(a)  $F1$  的收敛曲线



(b)  $F6$  的收敛曲线



(c)  $F8$  的收敛曲线



(d)  $F10$  的收敛曲线

图 2  $F1$ 、 $F6$ 、 $F8$ 、 $F10$  在不同算法中的收敛曲线

Fig. 2 Convergence curves of  $F1$ 、 $F6$ 、 $F8$ 、 $F10$  in different algorithms

表 2 给出了每个测试函数在不同算法下求出的最优解和均值。通过表 2 可知,改进后的算法在函数  $F6$  和  $F8$  上能够找到最优解,虽然算法 IPSOGSA 在函数  $F8$  上也能找到最优值,但是结合图 2(c) 可知,IGSA 比 IPSOGSA 更早找到最优值。对于其他函数来说,IGSA 的寻优精度也有了很大的提升,多次寻优的最优值和均值都高出其他算法多个数量级;对比基本的 PSO 和 GSA 算法,每个函数都高出十几个数量级,有些甚至高出三十个数量级;对比 IPSOGSA,IGSA 在函数  $F1$  上找到最优解的平均值提高 13 个数量级,在  $F3$ 、 $F9$  和  $F10$  上提升了 12 个数量级,在剩下的函数中均提升了 7 个数量级。

因此,改进后的 IGSA 寻优能力有了较大的增强,寻优精度有了很大的提升,收敛速度稳定,这充

分说明了 IGSA 的可行性和优越性。

#### 4 结束语

本文提出的 IGSA 算法通过在质量的计算中引入随机因子,基于适应度值对力进行有选择的合成,赋予适应度值较优粒子对应的力更大的随机数,增强了算法的随机性,有利于算法能跳出局部最优,避免陷入早熟。同时在位置更新中引入参数,避免某些粒子过于随意地更新,保证其收敛性。最后通过 10 个测试函数,对比 GSA 算法、PSO 算法和 IPSOGSA 算法,IGSA 的寻优精度和收敛速度都有着明显的提升,表明了改进后的算法有更好的优化能力。

表 2 各算法在  $F1 \sim F10$  上的测试结果  
Table 2 Test results of four algorithms on  $F1 \sim F10$

函数名称	解情况	GSA	PSO	IPSOGSA	IGSA
F1	最优解	0.021 8	0.050 3	6.999 4e-16	2.888 7e-31
	均值	0.044 2	0.627 4	1.226 2e-15	8.174 5e-28
F2	最优解	0.481 8	1.642 8	1.068 2e-07	2.185 9e-15
	均值	0.563 2	4.060 6	1.209 1e-07	3.566 1e-14
F3	最优解	1.098 6e+03	0.165 3	2.756 0e-12	1.838 7e-27
	均值	2.154 2e+03	2.826 6e+03	7.098 9e-12	3.991 6e-24
F4	最优解	0.101 7	0.070 9	9.665 6e-09	1.761 4e-16
	均值	0.513 7	0.087 5	1.389 7e-08	7.527 0e-15
F5	最优解	0.038 4	0.141 9	9.971 7e-09	1.163 3e-16
	均值	0.046 8	2.276 0	1.393 9e-08	3.488 2e-15
F6	最优解	14.623 7	131.457 3	1.296 7e-13	0
	均值	20.584 9	203.339 0	2.377 4e-13	0
F7	最优解	0.094 2	2.948 3	5.543 2e-08	8.881 8e-16
	均值	0.132 2	4.107 6	1.066 2e-07	2.066 5e-14
F8	最优解	0.004 0	0.353 2	0	0
	均值	0.011 3	0.659 5	0	0
F9	最优解	7.833 5	57.588 9	8.966 8e-15	4.495 3e-30
	均值	17.406 2	214.813 2	4.453 4e-14	1.941 8e-26
F10	最优解	1.105 1e-05	7.023 7e-06	3.981 1e-16	1.382 3e-29
	均值	4.285 5e-04	1.792 4e-04	4.115 2e-14	1.531 0e-26

## 参考文献

- [1] RASHEDI E, NEZAMABADI-POUR H, SARYAZDI S. GSA: a gravitational search algorithm[J]. Information sciences, 2009, 179(13): 2232-2248.
- [2] RATHER S A, BALA P S. Lévy flight and chaos theory-based gravitational search algorithm for mechanical and structural engineering design optimization[J]. Open Computer Science, 2021, 11(1): 509-529.
- [3] RATHER S A, BALA P S. Hybridization of chaotic maps and gravitational search algorithm for constrained mechanical and civil engineering design frameworks: CGSA for mechanical and civil engineering design optimization[J]. International Journal of Applied Metaheuristic Computing (IJAMC), 2022, 13(1): 1-39.
- [4] THIAGARAJAN K, MANAPAKKAM ANANDAN M, STATECZNY A, et al. Satellite image classification using a hierarchical ensemble learning and correlation coefficient-based gravitational search algorithm[J]. Remote Sensing, 2021, 13(21): 4351.
- [5] KUMAR Y, VERMA S K, SHARMA S. An ensemble approach of improved quantum inspired gravitational search algorithm and hybrid deep neural networks for computational optimization[J]. International Journal of Modern Physics C, 2021, 32(8): 2150100.
- [6] CHEN G, QIN F, LONG H, et al. Fuzzy PID controller optimized by improved gravitational search algorithm for load frequency control in multi-area power system[J]. IAENG International Journal of Computer Science, 2022, 49(1): 125-139.
- [7] KHANESAR M A, LU J, SMITH T, et al. Electrical load prediction using interval type-2 Atanassov intuitionist fuzzy system: gravitational search algorithm tuning approach[J]. Energies, 2021, 14(12): 3591.
- [8] AMPONSAN A A, HAN F, OSEI-KWAKYE J, et al. An improved multi-leader comprehensive learning particle swarm optimisation based on gravitational search algorithm[J]. Connection Science, 2021, 33(4): 803-834.
- [9] WANG Y, GAO S, YU Y, et al. A gravitational search algorithm with hierarchy and distributed framework[J]. Knowledge-Based Systems, 2021, 218(2): 106877.
- [10] 傅文渊. 具有万有引力加速机理的布谷鸟搜索算法[J]. 软件学报, 2021, 32(5): 1480-1494.
- [11] 张维平, 任雪飞, 李国强, 等. 改进的万有引力搜索算法在函数优化中的应用[J]. 计算机应用, 2013, 33(5): 1317-1320.
- [12] 马力, 刘丽涛. 万有引力搜索算法的分析与改进[J]. 微电子学与计算机, 2015, 32(9): 76-80.
- [13] 刘小刚, 欧阳自根. 改进万有引力搜索算法在函数优化中的应用[J]. 沈阳工业大学学报, 2021, 43(2): 193-197.
- [14] 邢宇浩. 万有引力搜索算法的改进与应用[D]. 郑州: 河南大学, 2018.
- [15] 陈博文, 邹海. 总结性自适应变异的粒子群算法[J]. 计算机工程与应用, 2022, 58(8): 67-75.