

改进JAYA算法求解工程设计优化问题

刘景森^{1,2}, 杨 杰², 李 煜³

(1. 河南大学智能网络系统研究所, 河南开封 475004; 2. 河南大学软件学院, 河南开封 475004;
3. 河南大学管理科学与工程研究所, 河南开封 475004)

摘 要: 为了更好求解工程设计约束优化问题, 进一步提升JAYA算法的全局寻优和应用能力, 提出一种基于多角色差异进化策略的改进JAYA算法. 首先引入余弦相似度策略, 通过旋转变换算子和非均匀变异算子对与最优个体余弦相似度较高的个体位置进行处理, 不仅加快了算法的收敛速度, 而且丰富了种群的多样性; 然后在个体位置更新中采用多角色策略, 并引入共生策略和柯西变异机制, 有效平衡和较好满足了算法在不同迭代时期对探索和挖掘能力的不同需求, 进而改善了算法的优化性能; 最后引入小孔成像反向学习策略, 则扩大了算法的搜索范围, 进一步提高了算法的收敛性和精度. 通过对10个复杂标准测试函数进行的多维度、多算法函数极值优化, 以及对5个CEC2020中描述的更具挑战性的复杂工程设计问题的优化求解, 都清楚地表明改进后算法的寻优精度、收敛性能、求解稳定性及对不同问题的适用性和有效性均有显著提升, 在求解工程设计优化问题上有较为明显的优势.

关键词: JAYA算法; 余弦相似度; 共生策略; 函数极值优化; 工程设计约束优化

基金项目: 河南省重点研发与推广专项(No.222102210065); 国家自然科学基金(No.71601071)

中图分类号: TP301.6

文献标识码: A

文章编号: 0372-2112(2023)09-2469-12

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20211446

Solving Engineering Optimization Design Problems Based on Improved JAYA Algorithm

LIU Jing-sen^{1,2}, YANG Jie², LI Yu³

(1. Institute of Intelligent Networks System, Henan University, Kaifeng, Henan 475004, China;

2. College of Software, Henan University, Kaifeng, Henan 475004, China;

3. Institute of Management Science and Engineering, Henan University, Kaifeng, Henan 475004, China)

Abstract: In order to better solve the engineering design constrained optimization problem and further improve the global optimization and application ability of the JAYA algorithm, the JAYA algorithm based on multi role differential evolution strategy is proposed. Firstly, the cosine similarity strategy is introduced, the individual positions with high cosine similarity to the optimal individual are processed by rotation change operator and non-uniform mutation operator, which not only accelerates the convergence speed of the algorithm, but also enriches the diversity of the population; Then the multi role strategy is introduced to the individual location updating, and the symbiosis strategy and cauchy mutation mechanism are introduced to effectively balance and better meet the different needs of the algorithm for exploration and mining ability in different iterative periods, which improves the optimization performance of the algorithm; Finally, the pinhole-imaging opposition-based learning strategy is introduced, which expands the search range of the algorithm and further improves the convergence and accuracy of the algorithm. Through the simulation experiment of function extremum optimization of the multi algorithms on multiple dimensions of the 10 complex benchmark test functions and the optimization of 5 more challenging complex engineering design problems described in CEC2020, the test results clearly show that the proposed algorithm has significantly better optimization accuracy, convergence performance, solution stability, applicability and effectiveness to different problems, and it has obvious advantages in solving engineering design optimization problems.

Key words: JAYA algorithm; cosine similarity; symbiosis strategy; function extremum optimization; Engineering constrained optimization

Foundation Item(s): Key R&D and Promotion Projects of Henan Province (No.222102210065); National Natural Science Foundation of China (No.71601071)

1 引言

在现实世界中,全局优化问题无处不在,而对于高维非线性复杂优化问题,传统的精确计算优化方法面临着巨大的困难和挑战.受特定自然现象启发提出的群智能优化算法却对这类问题有着较好的求解效果,得到了诸多学者的青睐和研究.如:受鸟群捕食过程启发提出的粒子群算法^[1,2],受正余弦函数数学模型启发提出的正余弦算法^[3],受座头鲸捕食行为启发提出的鲸鱼算法^[4],受樽海鞘在海洋中游弋和觅食行为启发提出的樽海鞘群算法^[5],受基于量子的原子模型启发提出的原子轨道搜索算法^[6]等等.这些智能优化算法的不断提出、改进和优胜劣汰,为解决大规模复杂问题提供了新的思路和方法,目前已在现实世界的各个领域得到广泛应用.

函数极值优化问题是测试算法寻优性能的主要方法,而工程设计约束优化问题则是智能优化算法的一个重要应用领域和研究热点.智能优化算法能在可接受时间内求得问题的最优或近似解,为复杂工程设计问题的求解提供了实用且有效的方法.如:汪逸晖等^[7]提出一种引入动态感知概率、莱维飞行策略以及变异更新机制的改进乌鸦搜索算法;Gupta等^[8]提出一种基于反向学习的混合自适应正余弦算法;刘景森等^[9]提出一种基于分段式随机惯性权重和最优反馈机制的鲸鱼优化算法;Salgotra等^[10]提出一种自适应樽海鞘群算法;Zhang等^[11]提出一种基于神经网络算法和教与学优化算法的有效混合方法;石建平^[12]提出一种用于解决约束优化问题的改进果蝇优化算法.这些改进算法均被应用于求解工程设计约束优化问题,并取得了良好效果,但求解的问题多为几个经典老问题,种类较少且比较简单,仍需探索求解能力更强、寻优精度更高、稳定性和普适性更好的算法.

JAYA算法^[13]是2016年由Rao提出的一种新型启发式智能优化算法,该算法结构简单、易于实现,且具有趋优避差的导向性特征,对许多问题有着较好的寻优性能,成为最近几年进化计算领域重要的研究和改进算法之一,目前已在微通道散热器的尺寸优化、多区域互联电力系统、文本聚类、柔性车间调度、固体氧化物燃料电池参数优化等多个领域得到成功应用.

不过,JAYA算法与其他基础性智能优化算法类似,其本身也存在着容易陷入局部极值、寻优精度有时不高和收敛速度较慢等缺点,在解决高维复杂优化问题时尤其明显.为此,许多学者针对JAYA算法的不足做出了相应改进.Yu等^[14]引入自适应惯性权重、基于经验的学习策略和混沌精英学习方法,提高了JAYA算法的寻优精度和稳定性.Ravipudi等^[15]引入logistic混沌映射策略,有效提高了JAYA算法的收敛速度,使算

法能够更好地在搜索空间中探索而不易陷入局部极值.Pradhan等^[16]分别引入线性惯性权重和基于模糊逻辑的非线性惯性权重,这两种惯性权重的引入提高了JAYA算法解决复杂问题的精度.Ingle等^[17]引入莱维飞行和贪婪选择策略,丰富了JAYA算法中种群的多样性,增强了算法的勘探能力.Zhang等^[18,19]引入包含三种不同操作算子的综合学习机制来更新个体位置,提高了JAYA算法的全局搜索能力.Iacca等^[20]引入莱维飞行用于产生JAYA算法位置更新公式中的随机数,提高了算法跳出局部极值的能力.Yang等^[21]引入改进的进化算子、线性种群缩减策略和广义反向学习机制,有效提升了JAYA算法的全局搜索能力.

这些改进使JAYA算法在各自应用领域的求解性能得到了提升,但早熟收敛、全局搜索和局部挖掘能力不平衡以及对于工程设计约束优化问题的求解效果与适应性欠佳等问题依然存在.为了继续提高JAYA算法的寻优能力和收敛性能,使之能够更广泛、更优越地求解工程设计约束优化问题,本文提出一种基于多角色差异进化的改进JAYA算法(Modified JAYA algorithm, M-JAYA).首先引入余弦相似度策略,对与最优个体余弦相似度较高的个体进行更新,不仅加快了算法的收敛速度,而且丰富了种群的多样性;然后在个体位置更新中采用多角色策略,并引入共生进化策略和柯西变异机制进行不同更新,有效平衡和较好满足了算法在不同迭代时期对探索和挖掘能力的不同需求;最后引入小孔成像反向学习策略,则进一步增强算法搜索的变化性,提高算法的收敛性和精度.通过对10个复杂标准测试函数进行的多维度、多算法函数极值优化对比测试表明,改进后算法M-JAYA的收敛性能、寻优精度和鲁棒性均有显著提升,求解效果相当出色.进而又通过对IEEE CEC2020中描述的减速器、机器人夹持器、齿轮系统、滚动轴承和静压推力轴承等5个更具挑战性的工程设计约束优化问题的求解,充分验证和显示了M-JAYA算法在处理不同类型工程优化设计问题时有着明显的优越性和适应性.

2 基本JAYA算法

Step1 设置算法初始参数:种群个体数量 N 、最大进化代数 Max_iter 、个体维度 D ,并在寻优范围内随机生成每个个体的初始位置 $x_i (i = 1, 2, \dots, N)$.

Step2 根据目标函数计算种群个体的适应度值 $f(x_i)$.

Step3 根据种群中个体的适应度值 $f(x_i)$,找出最好和最差适应度值 f_{\min} 和 f_{\max} ,并记录其位置 x_{best} 和 x_{worst} .

Step4 由式(1)对个体每一维的位置进行更新.

$$x_i^j(t+1) = x_i^j(t) + r_1 \cdot (x_{\text{best}}^j(t) - |x_i^j(t)|) - r_2 \cdot (x_{\text{worst}}^j(t) - |x_i^j(t)|) \quad (1)$$

其中, $x_{\text{best}}^j(t)$ 是当前全局最优位置的第 j 维值, $x_{\text{worst}}^j(t)$ 为当前全局最差位置的第 j 维值, $x_i^j(t)$ 是当前代中第 i 个个体在第 j 维的位置值, $x_i^j(t+1)$ 是更新后下一代中第 i 个个体在第 j 维的位置值, r_1, r_2 均为 $[0, 1]$ 之间均匀分布的随机数。

Step5 由目标函数 $f(x)$ 求出新个体的适应度值, 并对新旧解进行对比, 若新解较优, 则替换上代的个体位置, 否则保留原来的个体位置。

Step6 判断当前迭代次数 t 是否达到最大迭代次数, 若 $t \leq \text{Max_iter}$, 返回 Step2;

Step7 确定最终的最优值并输出。

3 改进算法 M-JAYA

3.1 引入余弦相似度策略

JAYA 算法与其他大多数基础性智能优化算法一样, 在算法迭代的中后期, 种群个体容易聚拢在当前最优个体周围, 甚至出现过度聚集和重叠现象, 致使算法很难保持种群的多样性, 可能陷入局部最优, 为此, 引入余弦相似度策略来解决这一问题。

余弦相似度反映了两个向量的方向一致性关系, 取值范围为 $[-1, 1]$, 其中 1 表示两个向量方向相同, -1 表示两个向量方向相反。本文引入的余弦相似度策略是通过构造当前代种群中个体位置与最优个体位置之间的向量, 从而更新那些余弦相似度较高的个体, 既有利于加快算法的收敛速度, 也丰富了种群的多样性。M-JAYA 的余弦相似度策略如下:

首先构造两个向量 \mathbf{a} 和 \mathbf{b} :

$$\begin{cases} \mathbf{a} = x_i(t) \\ \mathbf{b} = x_{\text{best}}(t) \end{cases} \quad (2)$$

其中, $x_i(t)$ 为当前代种群中个体位置, 满足 $i \in [1, N]$, $x_{\text{best}}(t)$ 为当前代全局最优个体位置。

定义 $\cos(\mathbf{a}, \mathbf{b})$ 为两个向量之间的相似度, 则个体位置之间的余弦相似度计算公式为:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| \times |\mathbf{b}|} \quad (3)$$

式中, 分子为向量 \mathbf{a} 和 \mathbf{b} 的内积, 分母为向量 \mathbf{a} 和 \mathbf{b} 模的乘积。

依次计算当前代种群中每个个体与最优个体位置之间的余弦相似度 $C = \cos(\mathbf{a}, \mathbf{b})$, 将与全局最优个体余弦相似度较高的个体筛选出来, 再与当代种群中个体的平均适应度值做比较, 对适应度值较差的个体位置融合状态转移算法^[22]中的旋转变换算子进行更新, 将适应度值较好的个体位置融合非均匀变异算子^[23], 以

扩大种群的搜索范围, 进而丰富种群的多样性。具体更新模型如算法 1 所示。

算法 1 余弦相似度策略阶段

输入: Max_iter, D, N

IF(0.5 < C < 1)

IF($f(x_i(t)) \geq f_{\text{avg}}$)

$$x_i(t+1) = x_{\text{best}}(t) + \alpha \cdot \frac{1}{N \cdot \|x_{\text{best}}(t)\|_2} \cdot R_r \cdot x_{\text{best}}(t) \quad (4)$$

ELSE

$$x_i(t+1) = x_i(t) + \Delta(t, y) \quad (5)$$

END IF

END IF

输出: x_i

算法 1 中, $f(x_i(t))$ 为个体位置 $x_i(t)$ 的适应度值, $x_i(t+1)$ 为更新后下一代个体的位置, f_{avg} 为当代种群的平均适应度值, 即 $f_{\text{avg}} = \frac{1}{N} \cdot \sum_{i=1}^N f(x_i)$, R_r 为 $[-1, 1]$ 之间均匀分布的随机数, 即 $R_r = 2 \cdot \text{rand} - 1$, N 为种群中个体数量, $\|x_{\text{best}}(t)\|_2$ 为 $x_{\text{best}}(t)$ 的欧氏范数, 而对于旋转因子 α , 本文中由贝塔分布与下不完全 Γ 函数的逆函数融合产生, 其计算公式为:

$$\alpha = \text{gammaincinv}\left(\lambda, 1 - \frac{t}{\text{Max_iter}}\right) + B(n_1, n_2) \quad (6)$$

式中, 下不完全 Γ 函数 $\gamma(\lambda, m) = \int_0^\lambda e^{-x} x^{m-1} dx$, $\text{gammaincinv}(\lambda, m)$ 为下不完全 Γ 函数的逆函数, 其中 λ 为形状参数, 且 $\lambda > 0$, 用于决定函数曲线的形状, m 为逆尺参数, 用于决定函数曲线的陡峭程度。本文中 $\lambda = 0.1, m = 1 - \frac{t}{\text{Max_iter}}$, 此时, 下不完全 Γ 函数的逆函数具有在迭代前期接近线性下降, 迭代后期接近指数下降的特性。 $B(n_1, n_2)$ 为 $[0, 1]$ 之间服从贝塔分布的随机数, 其中 n_1, n_2 为形状参数, 当 $n_1 = 1, n_2 = 3$ 时, 贝塔分布呈现非线性递减趋势。

对于非均匀变异步长 $\Delta(t, y)$, 其计算公式如下:

$$\Delta(t, y) = y \cdot \left(1 - q^{\left(1 - \frac{t}{\text{Max_iter}}\right)^\beta}\right) \quad (7)$$

$$\begin{cases} U_{m+1} = 4 \cdot U_m \cdot (1 - U_m) \\ y = (2 \cdot U_{m+1} - 1) \cdot \mathbf{r}([1, D]) \end{cases} \quad (8)$$

其中, q 和 U_m 均为 $[0, 1]$ 之间均匀分布的随机数, $U_{m+1} = 4 \cdot U_m \cdot (1 - U_m)$ 为 logistic 混沌映射, $\mathbf{r}([1, D])$ 为服从高斯分布的 $1 \times D$ 非零随机对角矩阵, β 为系统参数, 决定变异计算的非均匀程度, 通常取值为 $\beta = 2^{[23]}$ 。

3.2 引入多角色共生柯西变异

基本 JAYA 算法采用统一的位置更新公式,对于一些复杂多极值优化问题,由于缺少全局探索和局部挖掘能力的动态平衡机制,导致算法前期的全局搜索有时不够充分,容易陷入局部极值,而后期在最优解附近的局部精细挖掘能力不强,造成算法有时寻优精度不高和收敛速度较慢的情况.为此,本文受群体角色分工机制的启发,在保持 JAYA 算法原有种群规模不变的条件下,按个体适应度值从小到大进行排序,把种群均分为领导者、跟随者和链接者三个子种群.领导者种群执行基本 JAYA 算法的进化策略,跟随者种群执行融合共生进化策略^[24],链接者种群执行柯西变异进化策略.通过使用三种不同的进化策略,可以进一步平衡 JAYA 算法的探索和开发能力,进而改善算法的优化性能.具体数学模型如算法 2.

算法 2 多角色共生柯西变异阶段

输入: Max_iter, D, N

IF $(i \leq \frac{1}{3} \cdot N)$

$$x'_i(t+1) = x'_i(t) + r_1 \cdot (x'_{\text{best}}(t) - |x'_i(t)|) - r_2 \cdot (x'_{\text{worst}}(t) - |x'_i(t)|)$$

ELSE IF $(i \leq \frac{2}{3} \cdot N)$

$$x_i(t+1) = x_i(t) + |\text{Levy}(D)| \cdot (x_{\text{best}}(t) - M \cdot Z) \quad (9)$$

ELSE

$$x_i(t+1) = x_i(t) + \text{cauchy} \cdot (x_{\text{best}}(t) - x_r(t)) \quad (10)$$

END IF

输出: x_i

式(9)中, $M = \frac{1}{2} \cdot (x_i(t) + x_{i-1}(t))$ 为共生量,表示种群中第 i 和 $i-1$ 个个体位置的关系; Z 为受益因子,表示种群中第 i 和 $i-1$ 个个体位置相互作用时所获得的利益,通常取值为 $Z = 1$ ^[24]; $\text{Levy}(D)$ 为莱维分布随机数构成的 D 维矩阵.

式(10)中, $x_r(t)$ 为当前代种群中随机个体,且 $r \in [1, N]$, cauchy 为柯西算子,柯西分布的概率密度函数具有肥尾、低峰的特征,因此从概率上讲,柯西分布可以产生更多样化的个体和涵盖更广泛的搜索空间.

3.3 引入小孔成像反向学习策略

针对基本 JAYA 算法在求解高维复杂函数时容易陷入局部极值的缺点,本文引入基于光学定律的小孔成像反向学习策略.假设在某一空间中,有一束高度为 h 的火焰 p 在 X 轴上的投影为 $x_{\text{best}}(t)$ (全局最优个体位置),坐标轴的上下限分别为 X_{max} 和 X_{min} ,在基点 o 上放置一个带小孔的中间屏,火焰通过小孔在墙上得到一

个高度为 h' 的倒像 P' ,此时在 X 轴上得到一个通过小孔成像产生的反向点 $x'_{\text{best}}(t)$ (全局最优个体位置的反向解),由小孔成像原理和相似三角形易得出如下公式:

$$\frac{(X_{\text{max}} + X_{\text{min}})/2 - x_{\text{best}}(t)}{x'_{\text{best}}(t) - (X_{\text{max}} + X_{\text{min}})/2} = \frac{h}{h'} \quad (11)$$

令 $\frac{h}{h'} = k$,通过变换得到 $x'_{\text{best}}(t)$,其表达式如下:

$$x'_{\text{best}}(t) = \frac{(X_{\text{max}} + X_{\text{min}})}{2} + \frac{(X_{\text{max}} + X_{\text{min}})}{2 \cdot k} - \frac{x_{\text{best}}(t)}{k} \quad (12)$$

分析式(12)可知,当 $k=1$ 时,小孔成像反向学习为一般反向学习策略,通过改变墙与带小孔板的距离来调整调节因子 k ,进而可以得到更好的个体位置,本文采用了文献[25]中的取值 $k=12000$,函数优化效果较好.通过对全局最优个体位置进行小孔成像反向学习,再根据目标函数计算该新解的适应度值,若新解的质量优于之前的解,则保留小孔成像反向学习后的新解;反之,则保留之前的解.这有利于扩大算法的搜索范围,进而改善算法的性能,提高优化效率.

3.4 M-JAYA 算法流程

M-JAYA 算法描述如算法 3.

算法 3 M-JAYA

输入: Max_iter, D, N 适应度函数 $f(x)$

初始化种群 $x_i, i = (1, 2, \dots, N)$

初始化各参数 Max_iter, D, N

计算种群中个体的适应度值 $f(x_i)$

找出当前全局最优个体位置 x_{best}

$t = 1$

WHILE $(t \leq \text{Max_iter})$

计算当前种群中所有个体的平均适应度值 f_{avg}

FOR $i = 1:N$

根据式(2)和(3)对全局最优个体位置 x_{best} 和当前个体位置 $x_i(t)$ 计算余弦相似度 C

IF $(0.5 < C < 1)$

IF $(f(x_i(t)) \geq f_{\text{avg}})$

由式(6)计算旋转因子 α

由式(4)对个体的位置进行更新

ELSE

由式(7)和(8)计算非均匀变异步长 $\Delta(t, y)$

由式(5)对个体的位置进行更新

END IF

END IF

对更新后个体每一维进行边界处理

计算更新后个体的适应度值

IF(更新后个体适应度值 < 原个体适应度值 $f(x_i)$)

```

 $x_i$  =更新后个体位置
END IF
END FOR  $i$ 
根据个体适应度值对种群个体进行排序,并找出最好和最差个体的
位置 $x_{best}$ 和 $x_{worst}$ 
FOR  $i = 1:N$ 
IF ( $i \leq \frac{1}{3} \cdot N$ )
FOR  $j = 1:D$ 
生成 $[0, 1]$ 上的随机数 $r_1, r_2$ 
由式(1)对个体第 $j$ 维的位置进行更新
END FOR  $j$ 
ELSE IF ( $i \leq \frac{2}{3} \cdot N$ )
由式(9)对个体的位置进行更新
ELSE
由式(10)对个体的位置进行更新
END IF
对更新后个体每一维进行边界处理
计算更新后个体适应度值
IF(更新后个体适应度值 $<f(x_i)$ )
 $x_i$  =更新后个体位置
END IF
IF(更新后个体适应度值 $<$ 全局最优个体适应度值 $f(x_{best})$ )
 $x_{best}$  =更新后个体位置
END IF
END FOR  $i$ 
由式(12)对全局最优个体位置 $x_{best}$ 进行小孔成像反向学习,产生反
向解 $x'_{best}$ 
对产生的反向解 $x'_{best}$ 的每一维进行边界处理
计算 $x'_{best}$ 的适应度值 $f(x'_{best})$ 
IF ( $f(x'_{best}) < f(x_{best})$ )
 $x_{best} = x'_{best}$ 
END IF
 $t = t + 1$ 
END WHILE
输出:根据新的适应度值比较并输出最优解位置
    
```

4 仿真实验

为了全面检验本文算法 M-JAYA 的寻优能力,将 M-JAYA 与基本 JAYA 算法^[13]、Improved JAYA Optimization Alogorithm (IJAYA)^[14]、Comprehensive Learning Jaya Algorithm (CLJAYA)^[18,19]、正余弦算法(SCA)^[3]和

基于动态分级策略的改进正余弦算法(DSCA)^[26]共6种算法,分别在复杂标准测试函数和工程设计约束优化问题上进行优化测试.

为了保证实验的公平性与客观性,6种对比算法在相同条件下独立运行50次,种群大小均为30,最大进化代数 Max_iter = 1 000. 算法参数设置方面,4种 JAYA 类算法无需另设参数,而 SCA 和 DSCA 算法中只需设置用于控制步长的迭代因子初始值 $a=2$,这个取值与各自算法的原文献和源代码相同.

4.1 标准测试函数优化实验

4.1.1 10个标准测试函数

实验测试中,本文使用的10个典型复杂标准测试函数如表1所示,其中 $F_1(x) \sim F_2(x)$ 为连续单峰函数用于验证算法的收敛速度与寻优精度; $F_3(x) \sim F_{10}(x)$ 为复杂多峰函数,用来测试算法跳出局部最优的能力和全局搜索能力.

表1 10个基本测试函数

编号	函数名	取值范围	最优值
$F_1(x)$	Axis parallel hyperellipsoid	$[-5.12, 5.12]$	0
$F_2(x)$	Rotated hyper-ellipsoid	$[-65, 65]$	0
$F_3(x)$	Expansion of F10	$[-100, 100]$	0
$F_4(x)$	Michalewicz	$[0, \pi]$	-0.801 3
$F_5(x)$	Powell	$[-10, 10]$	0
$F_6(x)$	Salomon	$[-100, 100]$	0
$F_7(x)$	Discus	$[-100, 100]$	0
$F_8(x)$	Alpine	$[-10, 10]$	0
$F_9(x)$	Penalized2	$[-50, 50]$	0
$F_{10}(x)$	Schaffer F7	$[-100, 100]$	0

4.1.2 寻优精度分析

将测试函数 $F_1(x) \sim F_{10}(x)$ 的维度分别设置为 $D = 10/50/100$ 进行测试,表2统计了10维、50维、100维下各算法独立运行50次得到的寻优结果的最佳值、平均值和方差.

由表2的统计数据可以看出,除个别情况外,本文算法 M-JAYA 在各函数不同维度下的求解精度和稳定性均明显优于其他5种对比算法,且在全部10个函数的所有维度上,50次寻优结果的最佳值和平均值皆优于基本 JAYA 算法,显示出本文算法对于 JAYA 机制改进的显著有效性.

表2 6种算法在固定迭代次数下的寻优结果比较

函数	算法	$D=10$			$D=50$			$D=100$		
		最佳值	平均值	方差	最佳值	平均值	方差	最佳值	平均值	方差
$F_1(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	2.56E-05	1.22E-04	3.40E-09	6.31E+01	1.48E+02	6.78E+03	3.76E+02	7.36E+02	6.76E+04
	CLJAYA	3.73E-45	7.79E-37	2.85E-71	2.57E-04	5.21E+00	1.16E+02	1.58E-01	7.48E+01	8.75E+03

续表

函数	算法	D=10			D=50			D=100		
		最佳值	平均值	方差	最佳值	平均值	方差	最佳值	平均值	方差
	JAYA	4.60E-31	2.19E-28	8.03E-55	1.48E+02	4.43E+02	4.06E+04	1.03E+03	2.07E+03	4.39E+05
	DSCA	1.32E-195	2.54E-158	3.22E-314	5.80E-144	1.10E-119	5.38E-237	1.59E-140	1.58E-106	1.24E-210
	SCA	2.15E-42	3.70E-30	6.47E-58	2.73E-04	3.78E+00	8.42E+01	6.58E+00	6.28E+02	2.85E+05
$F_2(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	9.98E-02	3.28E-01	4.76E-02	9.86E+04	3.01E+05	2.34E+10	9.42E+05	3.88E+06	4.94E+12
	CLJAYA	2.55E-39	8.22E-32	3.37E-61	2.40E-01	7.99E+03	1.97E+08	5.87E+03	4.09E+05	2.38E+11
	JAYA	7.88E-26	2.89E-23	2.15E-45	1.35E+05	1.18E+06	2.76E+11	5.07E+06	1.16E+07	9.02E+12
	DSCA	1.39E-226	5.74E-176	0	5.01E-278	8.42E-151	3.54E-299	1.13E-254	1.39E-137	9.67E-273
	SCA	1.57E-36	7.18E-25	2.43E-47	6.34E+00	2.00E+03	4.78E+06	3.45E+04	1.53E+06	1.86E+12
$F_3(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	1.91E+01	3.18E+01	2.93E+01	1.23E+02	2.02E+02	1.15E+03	2.66E+02	4.47E+02	4.18E+03
	CLJAYA	2.15E-11	5.23E-02	6.62E-02	5.88E-06	5.80E+00	4.36E+01	2.02E-01	1.66E+01	2.20E+02
	JAYA	1.23E-01	1.14E+01	1.02E+02	1.55E+02	2.33E+02	1.46E+03	3.43E+02	4.83E+02	4.44E+03
	DSCA	2.31E-71	2.18E-49	1.14E-96	0	2.20E-42	2.43E-82	0	9.47E-44	4.37E-85
	SCA	1.98E-11	5.67E-08	9.46E-15	1.53E-01	8.34E+00	1.15E+02	6.64E+00	5.71E+01	2.11E+03
$F_4(x)$	M-JAYA	-9.62E+00	-8.22E+00	7.40E-01	-4.95E+01	-4.80E+01	2.81E+00	-9.84E+01	-9.50E+01	7.72E+00
	IJAYA	-8.68E+00	-6.99E+00	5.03E-01	-1.87E+01	-1.67E+01	9.18E-01	-2.94E+01	-2.73E+01	1.31E+00
	CLJAYA	-8.79E+00	-7.50E+00	4.13E-01	-2.57E+01	-2.23E+01	1.99E+00	-4.09E+01	-3.45E+01	1.04E+01
	JAYA	-8.11E+00	-6.51E+00	5.80E-01	-2.30E+01	-1.86E+01	3.23E+00	-3.99E+01	-3.10E+01	8.72E+00
	DSCA	-6.60E+00	-5.60E+00	1.61E-01	-2.01E+01	-1.79E+01	1.89E+00	-4.05E+01	-3.30E+01	2.07E+01
	SCA	-5.16E+00	-3.97E+00	2.62E-01	-1.35E+01	-1.08E+01	1.30E+00	-2.05E+01	-1.75E+01	2.13E+00
$F_5(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	1.29E-02	8.82E-02	2.10E-03	3.74E+04	1.36E+05	8.58E+09	9.75E+04	4.50E+05	9.91E+10
	CLJAYA	2.31E-18	6.88E-04	7.38E-06	8.64E-01	2.31E+03	7.33E+06	5.18E+01	9.83E+03	1.11E+08
	JAYA	1.94E-08	1.53E-07	2.35E-14	3.90E+03	1.47E+04	1.31E+08	2.02E+04	5.89E+04	1.26E+09
	DSCA	1.34E-183	7.33E-158	2.35E-313	4.24E-141	5.90E-119	9.36E-236	1.88E-131	2.30E-107	1.28E-212
	SCA	6.98E-27	4.08E-10	7.38E-18	8.95E+00	3.95E+03	1.06E+08	4.52E+03	1.35E+05	1.04E+10
$F_6(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	1.15E-01	3.49E-01	1.13E-02	6.78E+01	2.29E+02	6.49E+03	1.51E+02	7.44E+02	1.58E+05
	CLJAYA	3.29E-09	4.59E-04	4.36E-06	3.11E-03	9.96E+00	2.23E+02	7.96E-02	7.32E+01	7.17E+03
	JAYA	1.39E-03	8.34E-03	9.39E-05	1.79E+02	7.27E+02	8.81E+04	7.25E+02	1.74E+03	2.50E+05
	DSCA	2.55E-284	3.56E-178	0	9.88E-324	1.36E-158	9.18E-315	6.81E-303	2.97E-133	4.41E-264
	SCA	2.09E-13	2.85E-07	1.75E-12	2.96E-01	1.04E+01	3.64E+02	1.57E+01	5.01E+02	1.65E+05
$F_7(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	3.98E-01	2.01E+02	2.00E+06	8.08E+03	3.35E+04	1.07E+08	1.35E+04	9.11E+04	5.76E+08
	CLJAYA	3.74E-40	2.78E-33	1.98E-64	4.43E-05	2.22E+01	2.75E+03	3.73E-02	3.01E+02	2.45E+05
	JAYA	7.04E-26	1.74E-23	1.52E-45	9.40E+02	1.87E+04	9.14E+07	7.56E+03	5.06E+04	4.17E+08
	DSCA	1.45E-290	3.84E-192	0	0	4.42E-182	0	0	1.19E-190	0
	SCA	9.03E-36	7.97E-26	1.62E-49	1.60E-02	7.54E+01	1.86E+04	1.51E+02	4.63E+03	2.22E+07
$F_8(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	1.59E+00	3.30E+00	4.83E-01	3.01E+01	4.02E+01	2.82E+01	3.24E+01	6.47E+01	3.04E+02
	CLJAYA	2.01E-26	1.10E-04	6.00E-07	5.48E-15	6.30E-02	5.76E-02	3.12E-07	2.69E-01	1.18E-01
	JAYA	3.12E-12	1.14E-01	2.23E-01	9.54E+00	1.92E+01	5.37E+01	2.76E+01	5.89E+01	3.63E+02
	DSCA	1.03E-97	1.53E-80	1.17E-158	9.59E-81	2.04E-64	1.52E-126	1.57E-86	1.20E-63	4.59E-125
	SCA	1.26E-23	1.03E-04	4.95E-07	6.90E-03	1.55E+00	7.52E+00	2.26E-01	1.55E+01	9.39E+01

续表

函数	算法	D=10			D=50			D=100		
		最佳值	平均值	方差	最佳值	平均值	方差	最佳值	平均值	方差
$F_9(x)$	M-JAYA	1.35E-32	3.06E-03	2.90E-05	1.74E-08	3.60E-02	4.03E-03	6.97E-03	4.40E+00	4.99E+00
	IJAYA	3.91E-02	4.05E-01	8.60E-02	3.36E+07	1.28E+08	5.85E+15	1.35E+08	4.01E+08	6.86E+16
	CLJAYA	2.11E-03	9.20E-02	2.91E-02	2.47E+00	1.92E+05	1.67E+11	2.03E+01	8.80E+05	1.42E+12
	JAYA	6.73E-03	1.41E-01	5.73E-02	1.32E+02	1.57E+06	5.15E+12	1.31E+05	6.41E+06	4.26E+13
	DSCA	7.87E-03	1.00E-01	6.19E-03	3.73E+00	4.33E+00	3.05E-02	9.20E+00	9.44E+00	1.31E-02
	SCA	8.28E-02	2.69E-01	5.74E-03	1.17E+01	3.67E+06	3.72E+13	4.97E+07	3.05E+08	3.52E+16
$F_{10}(x)$	M-JAYA	0	0	0	0	0	0	0	0	0
	IJAYA	1.81E+00	3.17E+00	3.74E-01	2.94E+00	3.99E+00	3.12E-01	2.91E+00	4.36E+00	4.49E-01
	CLJAYA	8.25E-12	4.21E-03	3.76E-04	2.81E-03	1.41E-01	1.63E-02	2.19E-03	1.90E-01	3.47E-02
	JAYA	1.46E-02	5.85E-01	3.27E-01	2.59E+00	4.28E+00	6.23E-01	3.47E+00	4.82E+00	4.04E-01
	DSCA	1.47E-78	8.15E-51	3.27E-99	0	1.34E-49	5.48E-97	0	2.35E-48	2.64E-94
	SCA	1.95E-12	2.35E-08	8.14E-15	2.61E-03	1.72E-01	2.53E-02	6.83E-02	6.87E-01	3.35E-01

在 10 维条件下,对于 2 个单峰函数 $F_1(x)$ 和 $F_2(x)$, M-JAYA 的最佳值、平均值和方差均为理论最优值,求解效果明显优于其他 5 种对比算法. 对于 8 个复杂多峰函数 $F_3(x) \sim F_{10}(x)$, M-JAYA 的最佳值和平均值都优于其余 5 种算法,特别是对于函数 $F_3(x)$ 、 $F_5(x) \sim F_8(x)$ 和 $F_{10}(x)$, M-JAYA 在这 6 个函数上的最佳值、平均值和方差均为理论最优值,求解能力突出.

在 50 维和 100 维的高维条件下,6 种算法的求解精度都会随着维度的增加有所降低,但 M-JAYA 算法表现出优越的维度适应性,寻优精度仍优于其他 5 种算法. 对于 2 个单峰函数 $F_1(x)$ 和 $F_2(x)$, M-JAYA 在 50 维和 100 维下 50 次寻优结果的最佳值、平均值和方差均为理论最优值,求解能力依然十分出色. 对于 8 个多峰函数 $F_3(x) \sim F_{10}(x)$, M-JAYA 得到的最佳值、平均值也都优于其他 5 种算法,尤其是对于函数 $F_3(x)$ 、 $F_5(x) \sim F_8(x)$ 和 $F_{10}(x)$ 这 6 个函数, M-JAYA 在 50 维和 100 维下的最佳值、平均值和方差仍都是理论最优值,求解效果十分优越、稳定.

4.1.3 收敛曲线分析

一个算法性能的优劣,可以直观地通过收敛曲线展现出来,收敛曲线显示了算法在寻优过程中陷入局部最优的次数和收敛速度. 下面给出上述 10 个测试函数中 $F_1(x)$ 、 $F_7(x) \sim F_8(x)$ 和 $F_{10}(x)$ 共 4 个函数的收敛曲线图,其他函数的收敛曲线对比结果与之相似,不再冗赘列出.

图 1 至图 4 是 6 种算法在维度 $D=100$ 时求解函数 $F_1(x)$ 、 $F_7(x) \sim F_8(x)$ 和 $F_{10}(x)$ 的收敛曲线对比图.

图 1~4 清晰地展现了 M-JAYA、JAYA、CLJAYA、IJAYA、SCA 和 DSCA 算法在进化过程中适应度值的变化趋势. 从这些图中可以看出, M-JAYA 的收敛速度快于其他 5 种算法,且收敛曲线总体比较光滑,陷入局部

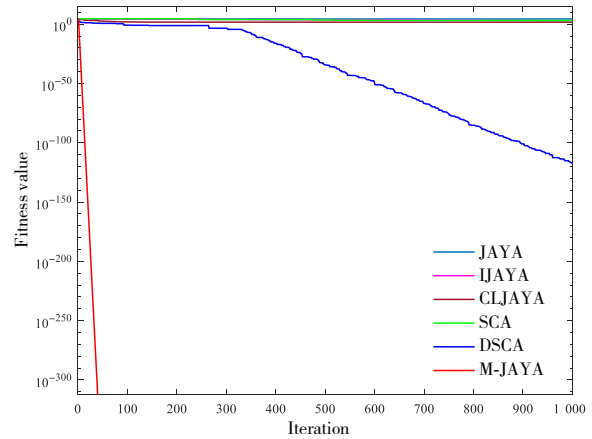


图 1 $F_1(x)$ 的收敛曲线

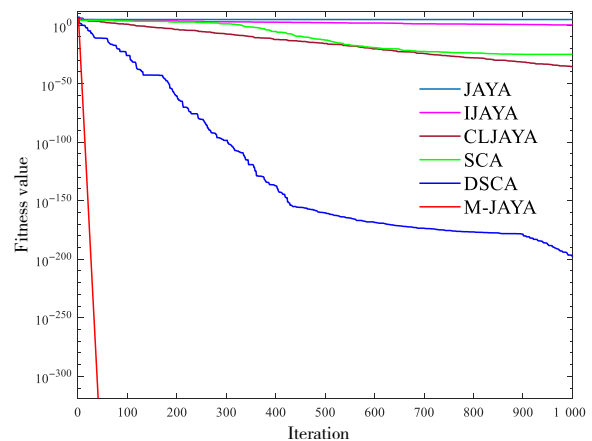


图 2 $F_7(x)$ 的收敛曲线

极值的次数较少,寻优能力更强.

4.1.4 实验结果的秩和检验统计分析

为了验证改进算法 M-JAYA 与其他对比算法在实验结果上的差异具有显著性,进一步评价算法的寻优性能,

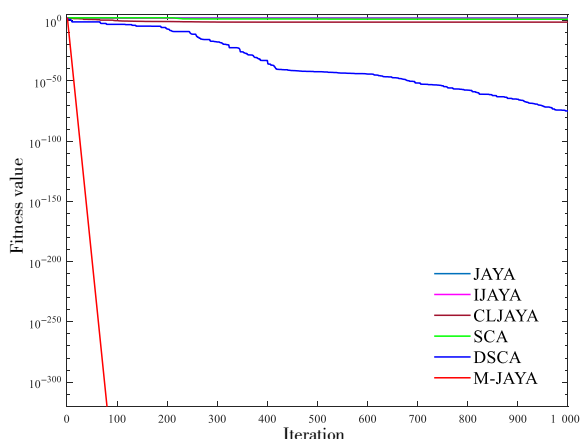


图3 $F_8(x)$ 的收敛曲线

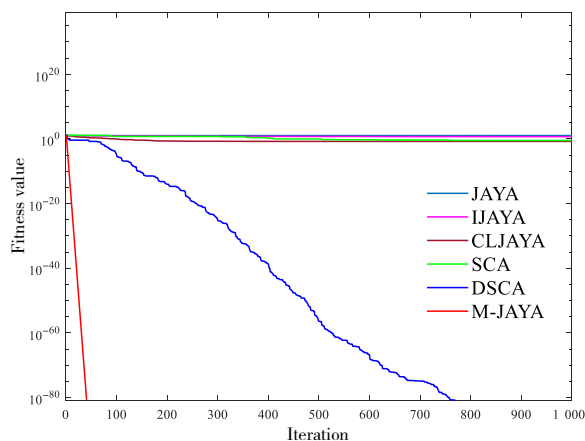


图4 $F_{10}(x)$ 的收敛曲线

采用非参数统计检验方法 Wilcoxon 秩和检验进行统计分析. 表3给出了表2中对于以上10个测试函数, M-JAYA 分别与其他对比算法在 $D=100$ 时的统计检验结果. 其中, 用符号“+”、“-”和“=”分别表示 M-JAYA 的寻优结果优于、劣于和相当于其他对比算法. 由文献[27]可知, 那些 $p < 0.05$ 的结果就可认为是拒绝零假设

具有显著性的有力验证.

从表3的统计检验结果来看, 本文算法 M-JAYA 与 JAYA、IJAYA、CLJAYA、SCA 和 DSCA 算法相比, 在全部10个测试函数上的检验 p 值都小于 0.05 且符号均为“+”, 拒绝零假设. 由此可见, M-JAYA 与其他5种对比算法的寻优结果之间具有显著性差异, 且 M-JAYA 显著更优.

4.2 工程设计约束优化问题的求解与分析

为了更进一步检验改进算法 M-JAYA 求解工程约束优化问题的能力, 将 M-JAYA 算法和上述5种代表性对比算法应用于 IEEE CEC2020 中描述的减速器、机器人夹持器、齿轮系统、滚动轴承和静压推力轴承等5个标准工程优化设计问题. 在求解这些问题时, 将每种算法独立运行50次, 得到设计结果的最佳值、平均值和方差, 便可作为评价各算法求解能力的指标.

4.2.1 求解减速器优化设计问题

减速器是飞机发动机和螺旋桨之间的一个齿轮箱, 属于非常复杂的工程件设计优化问题, 该设计问题的目的是在齿轮弯曲应力、表面应力、轴的横向挠度和轴的应力等约束条件下, 使减速器的重量最小化. 这个问题包含11个约束条件和7个设计变量, 设计变量分别是减速器的宽度 ($2.6 \leq x_1 \leq 3.6$)、齿模、小齿轮的齿数、转轴间第一轴的长度、转轴间第二轴的长度、第一轴的直径、第二轴的直径. 其数学模型如下.

目标函数:

$$f(x) = 0.785 4x_2^2 x_1 (14.933 4x_3 - 43.093 4 + 3.333 3x_3^2) + 0.785 4(x_5 x_7^2 + x_4 x_6^2) - 1.508 x_1 (x_7^2 + x_6^2) + 7.477 (x_7^3 + x_6^3)$$

约束条件:

$$g_1(x) = -x_1 x_2^2 x_3 + 27 \leq 0$$

$$g_2(x) = -x_1 x_2^2 x_3^2 + 397.5 \leq 0$$

表3 各算法求解10个测试函数的 Wilcoxon 秩和检验 p 值

函数	M-JAYA vs JAYA p -value win	M-JAYA vs IJAYA p -value win	M-JAYA vs CLJAYA p -value win	M-JAYA vs SCA p -value win	M-JAYA vs DSCA p -value win
$F_1(x)$	2.672 7E-04+	2.671 3E-04+	2.691 7E-13+	2.443 3E-03+	2.691 7E-13+
$F_2(x)$	2.691 7E-13+	2.691 7E-13+	9.892 8E-07+	1.638 7E-11+	2.691 7E-13+
$F_3(x)$	2.669 3E-04+	2.670 6E-04+	2.632 6E-04+	2.691 7E-13+	2.691 7E-13+
$F_4(x)$	2.691 7E-13+	2.691 7E-13+	2.632 6E-04+	2.691 7E-13+	2.691 7E-13+
$F_5(x)$	2.156 7E-04+	2.691 7E-13+	4.675 9E-04+	1.733 8E-05+	2.691 7E-13+
$F_6(x)$	2.673 4E-04+	2.672 7E-04+	4.809 2E-02+	1.336 3E-02+	2.691 7E-13+
$F_7(x)$	2.432 9E-04+	1.530 9E-08+	2.691 7E-13+	2.673 4E-04+	2.691 7E-13+
$F_8(x)$	7.178 9E-04+	5.189 8E-04+	3.075 3E-10+	2.623 1E-04+	2.691 7E-13+
$F_9(x)$	2.691 7E-13+	2.691 7E-13+	4.229 9E-05+	2.691 7E-13+	2.691 7E-13+
$F_{10}(x)$	1.085 5E-04+	8.063 2E-05+	2.691 7E-13+	1.836 4E-06+	2.691 7E-13+
+/-/=	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0

$$\begin{aligned}
 g_3(x) &= -x_2 x_6^4 x_3 x_4^{-3} + 1.93 \leq 0 \\
 g_4(x) &= -x_2 x_7^4 x_3 x_5^{-3} + 1.93 \leq 0 \\
 g_5(x) &= 10x_6^{-3} \sqrt{16.91 \times 10^6 + (745x_4 x_2^{-1} x_3^{-1})^2} - 1100 \leq 0 \\
 g_6(x) &= 10x_7^{-3} \sqrt{157.5 \times 10^6 + (745x_5 x_2^{-1} x_3^{-1})^2} - 850 \leq 0 \\
 g_7(x) &= x_2 x_3 - 40 \leq 0 \\
 g_8(x) &= -x_1 x_2^{-1} + 5 \leq 0 \\
 g_9(x) &= x_1 x_2^{-1} - 12 \leq 0 \\
 g_{10}(x) &= 1.5x_6 - x_4 + 1.9 \leq 0 \\
 g_{11}(x) &= 1.1x_7 - x_5 + 1.9 \leq 0
 \end{aligned}$$

$$\begin{aligned}
 g_3(x) &= Y_{\max} - y(x, 0) \leq 0 \\
 g_4(x) &= y(x, 0) - Y_G \leq 0 \\
 g_5(x) &= l^2 + e^2 - (a+b)^2 \leq 0 \\
 g_6(x) &= b^2 - (a-e)^2 - (l - Z_{\max})^2 \leq 0 \\
 g_7(x) &= Z_{\max} - l \leq 0
 \end{aligned}$$

其中, $a = \cos^{-1}\left(\frac{a^2 + g^2 - b^2}{2ag}\right) + \phi$, $g = \sqrt{e^2 + (z-l)^2}$,
 $Y_{\max} = 100$, $\beta = \cos^{-1}\left(\frac{b^2 + g^2 - a^2}{2bg}\right) - \phi$, $\phi = \tan^{-1}\left(\frac{e}{l-z}\right)$,
 $Y_G = 150$, $Z_{\max} = 100$, $P = 100$, $y(x, z) = 2(f + e + c \sin(\beta + \delta))$, $F_k = \frac{Pb \sin(\alpha + \beta)}{2c \cos(\alpha)}$, $Y_{\min} = 50$.

表4是M-JAYA算法和上述5种对比算法求解减速器设计问题得到寻优结果的最佳值、平均值和方差。从表中数据可以看出, M-JAYA算法的最佳值与IJAYA、CLJAYA、JAYA算法相同, 优于其余2种算法, 而平均值和方差则优于所有5种对比算法, 表现出更好的寻优精度和稳定性。而且M-JAYA的最佳值、平均值和方差均为理论最优值^[28], 求解效果十分出色。

表4 6种算法求解减速器设计问题的寻优结果比较

算法	最佳值	平均值	方差
M-JAYA	2.994 4E+03	2.994 4E+03	0
IJAYA	2.994 4E+03	2.994 5E+03	7.870 5E-04
CLJAYA	2.994 4E+03	3.001 5E+03	6.167 0E+02
JAYA	2.994 4E+03	3.011 6E+03	1.783 8E+03
DSCA	3.055 4E+03	3.126 0E+03	1.580 2E+03
SCA	3.044 8E+03	3.120 4E+03	1.228 6E+03

表5是M-JAYA和其他5种对比算法求解机器人夹持器设计问题的最佳值、平均值和方差。从表中数据可以看出, M-JAYA算法求解出的最佳值、平均值和方差都是6种算法中最好的, 求解性能明显优越。

表5 6种算法求解机器人夹持器设计问题的寻优结果比较

算法	最佳值	平均值	方差
M-JAYA	2.543 8E+00	2.657 1E+00	1.244 5E-02
IJAYA	2.545 1E+00	2.727 3E+00	4.113 4E-02
CLJAYA	2.544 3E+00	3.474 0E+00	8.564 4E-01
JAYA	2.544 8E+00	3.622 6E+00	1.717 2E+00
DSCA	3.580 2E+00	4.947 4E+00	3.160 7E+00
SCA	3.006 3E+00	4.545 2E+00	2.075 4E+00

4.2.2 求解机器人夹持器优化设计问题

夹持器作为机器人重要的末端执行器有着非常广泛的应用, 对它的优化设计是一个典型的非线性工程设计问题, 其设计目标是在夹持器末端位移范围内最大限度地减少最大和最小夹持力之间的差异, 以提高机器人夹持器的夹持精度, 实现其结构紧凑稳定、易于控制、活动轻便灵巧等优化效果。该问题有7个设计变量和7个约束条件, 这7个变量分别是第一根杠杆的长度 $a(10 \leq a \leq 150)$ 、第二根杠杆的长度 $b(10 \leq b \leq 150)$ 、第三根杠杆的长度 $c(10 \leq c \leq 150)$ 、轴承与气杆之间的距离 $e(0 \leq e \leq 50)$ 、气杆杆端与滑块之间的距离 $f(10 \leq f \leq 150)$ 、轴承与滑块之间的距离 $l(100 \leq l \leq 300)$ 、第二根杠杆 b 与第三根杠杆 c 之间的夹角 $\delta(1 \leq \delta \leq 3.14)$ 。其数学模型如下。

目标函数:

$$f(x) = -\min_z F_k(x, z) + \max_z F_k(x, z)$$

约束条件:

$$\begin{aligned}
 g_1(x) &= -Y_{\min} + y(x, Z_{\max}) \leq 0 \\
 g_2(x) &= -y(x, Z_{\max}) \leq 0
 \end{aligned}$$

4.2.3 求解齿轮系统设计问题

齿轮系统设计问题是一个著名的工程约束优化问题, 该问题中每个齿轮的最大齿轮数为60, 最小齿轮数为12。这个问题的设计目标是使复合齿轮系统设置的齿轮比 i_{tot} (即输出轴 ω_0 和输入轴 ω_i 的角速度之比) 与最佳齿轮比 i_{trg} 的差值最小, 以保证在复合齿轮系统设计过程中能够达到最佳比率。问题中复合齿轮系统使用两对齿轮 $x_1 - x_3$ 和 $x_2 - x_4$ 组装, 共包含4个变量和8个约束条件。其数学模型如下。

目标函数:

$$f(x) = (i_{\text{trg}} - i_{\text{tot}})^2 = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2$$

约束条件:

$$g_{1-4}(x) = 12 - x_i \leq 0 \quad g_{5,8}(x) = 60 - x \leq 0$$

其中, $i_{\text{tot}} = \frac{\omega_0}{\omega_i} = \frac{z_d z_b}{z_a z_f}$, z 是每个齿轮上的总齿数, $i_{\text{trg}} = \frac{1}{6.931}$ 。

表6统计了6种算法求解齿轮系统设计问题得到的最佳值、平均值和方差。由表中数据可知, 6种算法在求

解齿轮系统设计问题时,本文算法 M-JAYA 求得的最佳值、平均值和方差不仅是 6 种算法中最好的,而且都与文献[28]中给出的理论最优值相同,求解效果十分出色.

表 6 6 种算法求解齿轮系统设计问题的寻优结果比较

算法	最佳值	平均值	方差
M-JAYA	0	0	0
IJAYA	1.428 5E-20	2.358 6E-17	3.681 7E-33
CLJAYA	7.613 8E-22	3.842 3E-14	1.504 8E-26
JAYA	1.232 6E-32	5.217 6E-13	5.940 1E-24
DSCA	1.177 1E-14	2.429 3E-10	2.489 8E-19
SCA	1.266 0E-14	3.973 8E-10	7.010 9E-19

4.2.4 求解滚动轴承设计问题

滚动轴承设计是一个著名的复杂约束工程设计问题,其目的是使用 5 个设计变量来优化滚动轴承的承载能力. 该问题包括 5 个设计变量和 9 个约束条件,这 5 个变量分别是滚动体的直径 $D_b(0.15(D-d) \leq D_b \leq 0.45(D-d))$ 、保持架 $D_m(0.5(D+d) \leq D_m \leq 0.6(D+d))$ 、滚动体的数量 $Z(4 \leq Z \leq 50)$ 、内圈曲率系数 $f_i(0.515 \leq f_i \leq 0.6)$ 、外圈曲率系数 $f_o(0.515 \leq f_o \leq 0.6)$. 影响滚动轴承内部结构的参数分别为 $K_{D_{\min}}(0.4 \leq K_{D_{\min}} \leq 0.5)$ 、 $K_{D_{\max}}(0.6 \leq K_{D_{\max}} \leq 0.7)$ 、 $\delta(0.3 \leq \delta \leq 0.4)$ 、 $\eta(0.6 \leq \eta \leq 0.85)$ 、 $e(0.02 \leq e \leq 0.1)$. 具体数学模型如下.

目标函数:

$$f(x) = \begin{cases} f_c Z^{23} D_b^{1.8}, & \text{if } D_b \leq 25.4 \text{ mm} \\ 3.647 f_c Z^{23} D_b^{1.4}, & \text{if } D_b > 25.4 \text{ mm} \end{cases}$$

约束条件:

$$g_1(x) = Z - \frac{\phi_0}{2 \sin^{-1}(D_b/D_m)} - 1 \leq 0$$

$$g_2(x) = K_{D_{\min}}(D-d) - 2D_b \leq 0$$

$$g_3(x) = 2D_b - K_{D_{\max}}(D-d) \leq 0$$

$$g_4(x) = D_b - B_w \leq 0$$

$$g_5(x) = 0.5(D+d) - D_m \leq 0$$

$$g_6(x) = D_m - (0.5+e)(D+d) \leq 0$$

$$g_7(x) = \zeta D_b - 0.5(D - D_m - D_b) \leq 0$$

$$g_8(x) = 0.515 - f_i \leq 0$$

$$g_9(x) = 0.515 - f_o \leq 0$$

其中, $f_c = 37.91 \left\{ 1 + \left[1.04 \left(\frac{1-\gamma}{1+\gamma} \right)^{1.72} \left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)} \right)^{0.41} \right]^{10/3} \right\}^{-0.3}$,

$$\gamma = \frac{D_b \cos(\alpha)}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_i}{D_b}, T = D - d - 2D_b, D = 160, d = 90, B_w = 30, \phi_0 = 2\pi - 2 \times$$

$$\cos^{-1} \left[\frac{\left\{ (D-d)/2 - 3(T/4) \right\}^2 + \left\{ D/2 - T/4 - D_b \right\}^2 - \left\{ d/2 + T/4 \right\}^2}{2 \left\{ (D-d)/2 - 3(T/4) \right\} \left\{ D/2 - T/4 - D_b \right\}} \right]$$

表 7 统计了 6 种算法在求解滚动轴承设计问题得到的最佳值、平均值和方差. 从表中数据可以看出, 本文算法 M-JAYA 的最佳值与 CLJAYA、IJAYA、JAYA 算法相同, 且优于其余 2 种算法, 而平均值和方差则优于所有 5 种对比算法, 表现出更好的寻优精度和稳定性.

表 7 6 种算法求解滚动轴承设计问题的寻优结果比较

算法	最佳值	平均值	方差
M-JAYA	1.695 8E+04	1.696 2E+04	2.025 6E+02
IJAYA	1.695 8E+04	1.696 4E+04	9.836 7E+02
CLJAYA	1.695 8E+04	1.696 9E+04	5.399 7E+02
JAYA	1.695 8E+04	1.702 8E+04	3.779 0E+04
DSCA	1.706 6E+04	1.726 4E+04	2.811 8E+04
SCA	1.705 9E+04	1.729 7E+04	5.662 8E+04

4.2.5 静压推力轴承设计问题

静压推力轴承广泛应用于机床设备与高精密仪器中, 该设计问题的目的是最小化机械的摩擦功耗, 问题中包含 4 个设计变量和 7 个约束条件, 这 4 个设计变量分别为用油粘度 $\mu(1 \times 10^{-6} \leq \mu \leq 16 \times 10^{-6})$ 、轴承半径 $R(1 \leq R \leq 16)$ 、供油流量 $Q(1 \leq Q \leq 16)$ 、凹槽半径 $R_0(1 \leq R_0 \leq 16)$. 具体数学模型如下.

目标函数:

$$f(x) = \frac{QP_0}{0.7} + E_f$$

约束条件:

$$g_1(x) = 1000 - P_0 \leq 0, g_2(x) = W - 101000 \leq 0$$

$$g_3(x) = 5000 - \frac{W}{\pi(R^2 - R_0^2)} \leq 0$$

$$g_4(x) = 50 - P_0 \leq 0$$

$$g_5(x) = 0.001 - \frac{0.0307}{386.4P_0} \left(\frac{Q}{2\pi R h} \right) \leq 0$$

$$g_6(x) = R - R_0 \leq 0$$

$$g_7(x) = h - 0.001 \leq 0$$

其中, $W = \frac{\pi P_0 R^2 - R_0^2}{2 \ln(R/R_0)}$, $P_0 = \frac{6\mu Q}{\pi h^3} \ln\left(\frac{R}{R_0}\right)$, $E_f = 9336Q \times$

$$0.0307 \times 0.5 \Delta T, \Delta T = 2(10^P - 559.7),$$

$$P = \frac{\log_{10} \log_{10}(8.122 \times 10^6 \mu + 0.8) + 3.55}{10.04}$$

$$h = \left(\frac{2\pi \times 750}{60} \right)^2 \frac{2\pi\mu}{E_f} \left(\frac{R^4}{4} - \frac{R_0^4}{4} \right)$$

表 8 统计了 6 种算法求解静压推力轴承设计问题的最佳值、平均值和方差。从表中数据可以看出, M-JAYA 算法的最佳值、平均值和方差都是 6 种算法最好的, 表现出非常优越的求解性能。

表 8 6 种算法求解静压推力轴承设计问题的寻优结果比较

算法	最佳值	平均值	方差
M-JAYA	1.616 1E+03	1.772 4E+03	1.427 2E+04
IJAYA	1.843 6E+03	2.059 0E+03	2.026 2E+04
CLJAYA	1.616 2E+03	1.918 1E+03	4.107 0E+04
JAYA	1.702 6E+03	2.801 0E+03	1.000 4E+06
DSCA	1.969 5E+03	2.350 8E+03	3.145 9E+04
SCA	2.591 7E+03	3.392 6E+03	2.530 3E+05

以上求解结果和分析以及相关文献^[29,30]的求解结果都表明, 智能优化算法对于求解工程设计约束优化问题是有效的, 而本文所使用的代表性对比算法在求解工程约束优化问题时都具有良好的求解性能, 其中本文算法 M-JAYA 在面对不同类型工程设计问题时更是展现出较好的寻优精度和稳定性, 对于求解工程约束优化问题有着明显的优越性和适用性。

5 结论

本文提出一种基于多角色差异进化的改进 JAYA 算法 M-JAYA, 用于解决工程设计优化问题。该算法克服了基本 JAYA 算法寻优精度不高和收敛速度较慢等不足, 拓宽了 JAYA 算法的应用领域。改进算法通过引入余弦相似度策略, 对与最优个体余弦相似度较高的个体位置使用旋转变换算子和非均匀变异算子进行处理, 加快了算法的收敛速度, 丰富了种群的多样性; 在个体位置更新中采用多角色策略, 并引入共生策略和柯西变异机制, 有效平衡了算法在不同时期对探索和挖掘能力的不同需求, 改善了算法的优化性能; 引入小孔成像反向学习策略, 扩大了算法的搜索范围, 进一步提高了算法的收敛性和精度。将 6 种算法应用在 10 个标准测试函数和 5 个复杂工程设计问题上, 实验结果表明, 相较于其他代表性对比算法, 本文算法 M-JAYA 在寻优精度、收敛性能和算法稳定性等方面均有显著提升, 且有效验证了 M-JAYA 对于求解各类工程设计约束优化问题的可行性与优越性。在后续的研究中, 考虑将算法应用到更多领域的实际问题求解中, 进一步检验算法性能, 完善算法的改进机制, 拓展和提高算法的应用领域与能力。

参考文献

[1] KENNEDY J, EBERHART R. Particle swarm optimization [C]//Proceedings of ICNN'95 - International Conference on

Neural Networks. Piscataway: IEEE, 2002: 1942-1948.

- [2] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8 (3): 256-279.
- [3] MIRJALILI S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.
- [4] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [5] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.
- [6] AZIZI M. Atomic orbital search: A novel metaheuristic algorithm[J]. Applied Mathematical Modelling, 2021, 93: 657-683.
- [7] 汪逸晖, 高亮. 乌鸦搜索算法的改进及其在工程约束优化问题中的应用[J]. 计算机集成制造系统, 2021, 27(7): 1871-1883.
- WANG Y H, GAO L. Improvement of crow search algorithm and its application in engineering constrained optimization problems[J]. Computer Integrated Manufacturing Systems, 2021, 27(7): 1871-1883. (in Chinese)
- [8] GUPTA S, DEEP K. A hybrid self-adaptive sine cosine algorithm with opposition based learning[J]. Expert Systems With Applications, 2019, 119: 210-230.
- [9] 刘景森, 马义想, 李煜. 改进鲸鱼算法求解工程设计优化问题[J]. 计算机集成制造系统, 2021, 27(7): 1884-1897.
- LIU J S, MA Y X, LI Y. Improved whale algorithm for solving engineering design optimization problems[J]. Computer Integrated Manufacturing Systems, 2021, 27(7): 1884-1897. (in Chinese)
- [10] SALGOTRA R, SINGH U, SINGH S, et al. Self-adaptive salp swarm algorithm for engineering optimization problems[J]. Applied Mathematical Modelling, 2021, 89: 188-207.
- [11] ZHANG Y Y, JIN Z G, CHEN Y. Hybrid teaching—learning-based optimization and neural network algorithm for engineering design optimization problems[J]. Knowledge-Based Systems, 2020, 187: 104836.
- [12] 石建平, 李培生, 刘国平, 等. 求解约束优化问题的改进果蝇优化算法及其工程应用[J]. 控制与决策, 2021, 36(2): 314-324.
- SHI J P, LI P S, LIU G P, et al. Improved fruit fly optimization algorithm for solving constrained optimization prob-

- lems and engineering applications[J]. Control and Decision, 2021, 36(2): 314-324. (in Chinese)
- [13] VENKATA RAO R. JAYA: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems[J]. International Journal of Industrial Engineering Computations, 2016: 19-34.
- [14] YU K J, LIANG J J, QU B Y, et al. Parameters identification of photovoltaic models using an improved JAYA optimization algorithm[J]. Energy Conversion and Management, 2017, 150: 742-753.
- [15] RAVIPUDI J L, NEEBHA M. Synthesis of linear antenna arrays using JAYA, self-adaptive JAYA and chaotic JAYA algorithms[J]. AEU - International Journal of Electronics and Communications, 2018, 92: 54-63.
- [16] PRADHAN C, BHENDE C N. Online load frequency control in wind integrated power systems using modified JAYA optimization[J]. Engineering Applications of Artificial Intelligence, 2019, 77: 212-228.
- [17] INGLE K K, JATOTH D R K. An efficient JAYA algorithm with lévy flight for non-linear channel equalization [J]. Expert Systems With Applications, 2020, 145: 112970.
- [18] ZHANG Y Y, MA M D, JIN Z G. Comprehensive learning JAYA algorithm for parameter extraction of photovoltaic models[J]. Energy, 2020, 211: 118644.
- [19] ZHANG Y Y, JIN Z G. Comprehensive learning JAYA algorithm for engineering design optimization problems[J]. Journal of Intelligent Manufacturing, 2022, 33(5): 1229-1253.
- [20] IACCA G, DOS SANTOS V C Jr, VELOSO DE MELO V. An improved JAYA optimization algorithm with Lévy flight[J]. Expert Systems With Applications, 2021, 165: 113902.
- [21] YANG X, GONG W Y. Opposition-based JAYA with population reduction for parameter estimation of photovoltaic solar cells and modules[J]. Applied Soft Computing, 2021, 104: 107218.
- [22] ZHOU X J, TIAN J T, LONG J P, et al. A fast constrained state transition algorithm[J]. Neurocomputing, 2021, 455: 202-214.
- [23] ZHAO X C, GAO X S, HU Z C. Evolutionary programming based on non-uniform mutation[J]. Applied Mathematics and Computation, 2007, 192(1): 1-11.
- [24] ABDULLAHI M, NGADI M A, DISHING S I, et al. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment[J]. Journal of Network and Computer Applications, 2019, 133: 60-74.
- [25] 张达敏, 徐航, 王依柔, 等. 嵌入Circle映射和逐维小孔成像反向学习的鲸鱼优化算法[J]. 控制与决策, 2021, 36(5): 1173-1180.
- ZHANG D M, XU H, WANG Y R, et al. Whale optimization algorithm for embedded Circle mapping and onedimensional oppositional learning based small hole imaging[J]. Control and Decision, 2021, 36(5): 1173-1180. (in Chinese)
- [26] 魏锋涛, 张洋洋, 黎俊宇, 等. 基于动态分级策略的改进正余弦算法[J]. 系统工程与电子技术, 2021, 43(6): 1596-1605.
- WEI F T, ZHANG Y Y, LI J Y, et al. Improved sine cosine algorithm based on dynamic classification strategy[J]. Systems Engineering and Electronics, 2021, 43(6): 1596-1605. (in Chinese)
- [27] DERRAC J, GARCÍA S, MOLINA D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation, 2011, 1(1): 3-18.
- [28] KUMAR A, WU G H, ALI M Z, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results[J]. Swarm and Evolutionary Computation, 2020, 56: 100693.
- [29] LI Y, ZHAO Y R, LIU J S. Dimension by dimension dynamic sine cosine algorithm for global optimization problems[J]. Applied Soft Computing, 2021, 98: 106933.
- [30] KHISHE M, M. R MOVSAVI. Chimp optimization algorithm[J]. Expert Systems With Applications, 2020, 149: 113338.

作者简介



刘景森 男, 1968年生, 河南开封人, 博士, 教授, 研究方向为智能算法、优化控制和网络安全等.

E-mail: ljs@henu.edu.cn

杨杰 男, 1993年生, 山西忻州人, 硕士研究生, 研究方向为智能算法.

E-mail: yjehenu@163.com

李煜(通讯作者) 女, 1969年生, 河南开封人, 博士, 教授, 研究方向为智能算法和电子商务等.

E-mail: leey@henu.edu.cn