

李 崑, 钱 谦. 基于黏菌优化算法的机械传动行星轮系多目标优化设计[J]. 长安大学学报(自然科学版), 2024, 44(4): 149-160.
LI Kun, QIAN Qian. Multi-objective optimization design of mechanical transmission planetary gear train based on slime mold optimization algorithm[J]. Journal of Chang'an University (Natural Science Edition), 2024, 44(4): 149-160.

DOI: 10.19721/j.cnki.1671-8879.2024.04.014

基于黏菌优化算法的机械传动行星轮系多目标优化设计

李 崑^{1,2}, 钱 谦^{1,2}

(1. 昆明理工大学 信息工程与自动化学院, 云南 昆明 650500;
2. 昆明理工大学 云南省计算机技术应用重点实验室, 云南 昆明 650500)

摘 要: 为了优化机械传动中的关键部件行星轮系设计模型, 提出改进的黏菌优化算法。该算法通过加权聚合学习机制, 使黏菌个体在搜索空间中能够更好地学习和利用其他个体的优秀信息, 从而提高收敛速度和优化精度。将行星轮系的传动比、齿轮齿数、模数等关键参数作为优化变量, 以变量之间所满足的关系为约束条件, 以传动效率、体积、噪音等性能指标作为优化目标。通过构建合适的适应度函数, 将行星轮系设计模型优化问题转化为一个多目标优化问题, 并将该算法与 9 个对比算法在函数测试集和行星轮设计模型上进行试验验证。研究结果表明: 基于加权聚合学习机制的黏菌优化算法进行行星轮系设计优化效果显著, 具有收敛速度快、优化精度高、稳定性好等优点, 不仅能够在较短时间内找到全局最优解, 而且能够提供更加稳定和可靠的优化结果。

关键词: 机械工程; 行星轮系; 黏菌优化算法; 函数优化; 工程优化; 加权聚合学习

中图分类号: U463.21 **文献标志码:** A **文章编号:** 1671-8879(2024)04-0149-12

Multi-objective optimization design of mechanical transmission planetary gear train based on slime mold optimization algorithm

LI Kun^{1,2}, QIAN Qian^{1,2}

(1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, Yunnan, China; 2. Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming 650500, Yunnan, China)

Abstract: Amid at crucial component in mechanical transmission in mechanical transmission, a planetary gear system design optimization model based on improved slime mold optimization algorithm was proposed. The weighted aggregation learning mechanism was introduced, the algorithm enables slime mold individuals to better learn and utilize excellent information from other individuals in the search space, thereby accelerating convergence speed and improving optimization accuracy. Key parameters such as transmission ratio, gear tooth number, and

收稿日期: 2024-01-25

基金项目: 云南省基础研究计划项目(202101AT070082), 云南省计算机技术应用重点实验室开放基金项目

作者简介: 李 崑(1981-), 男, 北京市人, 工学硕士, E-mail: qianqian_yn@126.com。

通讯作者: 钱 谦(1981-), 男, 云南安宁人, 副教授, 工学博士, E-mail: qianqian_yn@126.com。

modulus of planetary gear systems was used as optimization variables, and the relationship between variables was taken as constraint conditions, and performance indicators such as transmission efficiency, volume, and noise was used as optimization objectives. By constructing appropriate fitness functions, the planetary gear train design optimization problem was transformed into a multi-objective optimization problem. And experimental verification on the function test set and planetary gear design through 9 comparative algorithms was conducted. The results show that the slime mold optimization algorithm based on the weighted aggregation learning mechanism achieves significant effects in planetary gear train design optimization. Compared with traditional optimization algorithms, this algorithm can not only find the global optimal solution in a shorter time but also provide more stable and reliable optimization results. The proposed algorithm provides a novel solution for the design optimization problem of planetary gear trains, and have advantages in fast convergence speed, high optimization accuracy and good stability. 6 tabs, 5 figs, 19 refs.

Key words: mechanical engineering; planetary gear system; slime mould algorithm; function optimization; engineering optimization; weighted aggregation learning

0 引 言

科学技术的进步和工程应用的深入,对机械传动中行星轮系的性能提出了更高要求。行星轮系作为一种复杂的机械传动机构,其设计涉及到多个参数的确定和优化,如齿轮齿数、模数、啮合角、变位系数等。这些参数不仅直接影响行星轮系的体积和传动效率,还关系到其噪音水平和使用寿命等。因此,深入研究行星轮系的设计优化问题,对于提升机械传动系统整体性能具有重要的理论价值和实践意义。

从算法角度来说,机械传动行星轮系设计优化是一个多约束、多目标、多变量的优化问题,需要综合考虑各种因素,以实现整体性能的最优化。智能算法中的黏菌算法(slime mould algorithm, SMA)^[1]以其独特的觅食行为和静脉网络形成机制,为机械传动行星轮系设计优化问题提供了新的思路。尽管 SMA 在解决行星轮系设计优化问题方面具有很大的潜力,但是需要在算法参数设置、适应度函数设计、约束处理、收敛性和稳定性等方面进行改进。

已有研究对 SMA 进行了改进和完善,如 Naik 等将反向学习机制引入 SMA,并利用当前位置较好的黏菌个体去替代其中一个随机选择的黏菌个体,提出了自适应对抗黏菌优化算法(adaptive opposition slime mould algorithm, AOSMA),增强了 SMA 的全局探索能力和局部开发能力,并改进了二者的平衡性^[2]。Liu 等在 SMA 的基础上引入单纯形策略,使用混沌机制取代了用于位置更新策略选择的随机数,提出了单纯形混沌黏菌优化算法(chaotic nelder-mead

slime mould algorithm, CNMSMA),强化了 SMA 的性能,并进一步在单二极管模型、双二极管模型和三二极管模型的 RTC France 太阳能电池和 PVM 752 GaAs 电池模型优化中验证了 CNMSMA 的有效性^[3]。Rizk-Allah 等^[4]在 SMA 中引入反向交叉策略以增加解的多样性、提高算法的探索能力,并融入混沌搜索策略以改善开发能力、避免过早收敛的问题,提出了基于混沌反向学习的黏菌优化算法(chaos-opposition-enhanced slime mould algorithm, CO-SMA),并在高海拔地区风力涡轮的设计问题上进行了验证。Zhao 等设计了结合扩散机制和关联策略的黏菌优化算法(diffusion mechanism and association strategy slime mould algorithm, DASMA),该算法将扩散机制引入到原初 SMA 中,增加了种群的多样性,能够更好地避免种群陷入局部最优,同时添加关联策略帮助 DASMA 更快地找到最优解^[5]。Hu 等^[6]为了提高 SMA 在多模态和混合函数中的性能,提出了一种带有分散觅食策略的黏菌优化算法(dispersed foraging slime mould algorithm, DFSMA),经过加利福尼亚大学尔湾分校(University of California-Irvine, UCI)的 12 个数据集评估,该算法不但提高了分类准确率,还减少了所选特征的数量,表明其在空间搜索和特征选择方面具有实用价值。贾鹤鸣等将 SMA 与算术优化算法(AOA)相结合,提出了一种融合随机反向学习策略的黏菌与算术混合优化算法(HSMAAOA),有效提高了 SMA 的收敛精度,并通过焊接梁的设计和压力容器的设计,验证了 HSMAAOA 在工程问题中的适用性和有效性^[7-8]。卢万杰等提出一种 Bernoulli

混沌初始化策略、动态非线性递减策略和融合麻雀预警机制与折射反向学习策略的改进黏菌算法(ISMA),利用变压器绝缘油中溶解气体数据,进行变压器故障诊断实验,验证了算法良好的工程实用性^[9]。除此之外,改进的黏菌优化算法还在许多工程问题中发挥着重要的作用,如无人机路径规划^[10]、田路分割^[11]和变压器故障识别^[12]等。

综上,现有的 SMA 及其改进算法在相当多的实际工程应用中取得了较好的效果,但是在行星轮系设计优化应用方面还存在一些问题,其整体优化效果有待提升。如 CNMSMA 引入了混沌映射机制,替换了控制种群演进过程中的随机参数,实现了种群探索与开发的平衡,但是在解决行星轮系设计中的多目标优化问题时效果并不明显。为此,本文提出一种基于加权聚合学习机制的黏菌优化算法(weighted aggregation slime mould algorithm, WASMA),意在提高行星轮系设计模型优化算法的收敛速度和跳出局部极值的能力,使行星轮系设计模型拥有更高的寻优精度。

1 行星轮系设计模型

行星轮系设计的核心目标在于提升其性能,具体包括但不限于提高传动效率、降低噪音和振动、减小体积和重量、增强耐用性和可靠性等^[13]。行星轮系设计的数学模型如下:

目标函数 $f(\vec{x})$ 为

$$\min f(\vec{x}) = \max |a_j - b_j| \quad j=1,2,3$$

式中: $a_1 = \frac{N_6}{N_4}$; $b_1 = 3.11$; $a_2 = \frac{N_6(N_1N_3 + N_2N_4)}{N_1N_3(N_6 - N_4)}$; $b_2 =$

1.84 ; $a_3 = \frac{N_2N_6}{N_1N_4}$; $b_3 = -3.11$; $N_1, N_2, N_3, N_4, N_5,$

N_6 均为齿数。

约束条件 $g_1(\vec{x}) \sim g_{10}(\vec{x})$ 为

$$g_1(\vec{x}) = m_2(N_6 + 2.5) \leq 220$$

$$g_2(\vec{x}) = m_1(N_1 + N_6) + m_2(N_1 + N_6) \leq 220$$

$$g_3(\vec{x}) = m_2(N_4 + N_5) + m_2(N_5 + 2) \leq 220$$

$$g_4(\vec{x}) = |m_1(N_1 + N_2) - m_2(N_6 + N_3)| \leq (m_1 + m_2)$$

$$g_5(\vec{x}) = (N_1 + N_6) \sin(\pi/P) - N_2 - 2 - d \geq 0$$

$$g_6(\vec{x}) = (N_6 - N_3) \sin(\pi/P) - N_3 - 2 - d \geq 0$$

$$g_7(\vec{x}) = (N_4 + N_5) \sin(\pi/P) - N_5 - 2 - d \geq 0$$

$$g_8(\vec{x}) = (N_6 - N_3)^2 + (N_4 + N_5)^2 - 2(N_6 -$$

$$N_3)(N_4 - N_5) \cos\left(\frac{2\pi}{P-\beta}\right) \geq (N_3 + N_5 + 2 + d)^2$$

$$g_9(\vec{x}) = N_6 - 2N_3 - N_4 - 4 - 2d \geq 0$$

$$g_{10}(\vec{x}) = N_6 - N_4 - 2N_5 - 4 - 2d \geq 0$$

$$h(\vec{x}) = \left(\frac{N_6 - N_4}{P}\right) = \text{integer}$$

$$\beta = \arccos \frac{[(N_6 - N_3)^2 + (N_4 + N_5)^2 - (N_3 + N_5)^2]}{2(N_6 - N_3)(N_4 + N_5)}$$

式中: $h(\vec{x})$ 、 d 和 β 均为计算参数; $d = 0.5, 17 \leq N_1 \leq 96, 14 \leq N_2 \leq 54, 14 \leq N_3 \leq 51, 17 \leq N_4 \leq 46, 14 \leq N_5 \leq 51, 48 \leq N_6 \leq 124$; integer 为取整函数; P 为齿轮数; m_1, m_2 为齿轮组件, m_1, m_2 取 $\{1.75, 2, 2.25, 2.5, 2.75, 3\}$ 。

行星轮系设计所追求的目标应当在于不断提高其功能,提高传动效率,增强耐用性和可靠性等,还在于降低机械结构的体积和重量,降低机械传动中的噪音和振动等。为了达到上述目的,就需要在理论上不断优化行星轮系设计模型。然而正如前文所述,行星轮系设计优化模型涉及的变量和约束条件较多,并且相互之间存在复杂的相互作用关系,需要通过优化算法找到最佳的设计组合。而行星轮系优化设计是一个多目标优化问题,需要同时考虑多个性能指标,并找到这些指标之间的平衡点。

2 加权聚合学习机制的黏菌优化算法

2.1 黏菌优化算法

SMA 是一种模拟黏菌移动和觅食行为的优化算法。黏菌的前端呈扇形,后面是相互连接的静脉网络,当静脉接近食物时,黏菌的生物振荡器就会产生扩散波来改变静脉中细胞质的流动,使黏菌向更好的食物移动,同时黏菌在寻找食物的过程中会分割一部分个体进行随机探索。其数学模型如下

$$X(t+1) = \begin{cases} \text{rand}(B_U - B_L) + B_L & \text{rand} < z \\ X_b(t) + \vec{v}_b[W X_A(t) - X_B(t)] & \text{rand} \geq z, r < p \\ \vec{v}_b X(t) & \text{rand} \geq z, r \geq p \end{cases} \quad (1)$$

式中: B_U, B_L 分别为搜索范围的上限和下限; $\text{rand}(\cdot)$ 和 r 为 $[0, 1]$ 区间的随机数; z 为随机分布的黏菌个体在黏菌总体中占比的参数; $X_b(t)$ 为目前适应度最高的个体位置; \vec{v}_b 为 $[-a, a]$ 内的随机数; W 为黏菌的权重系数; $X_A(t)$ 和 $X_B(t)$ 为从黏菌中随机选取的 2 个个体 A, B 的位置; \vec{v}_c 为绝对值从 1 到 0 递减的参数, $v_c \in [-1, 1]$; p 为决定黏菌位置更新方式的参数。

p 采用式(2)计算

$$p = \tanh |S(i) - F_D| \quad (2)$$

式中: $S(i)$ 为第 i 个黏菌个体 X_i 的适应度, $i=1,2,\dots,N$; F_D 为当前迭代下黏菌的最佳适应度。

a 值更新如下

$$a = \operatorname{arctanh}\left(-\frac{t}{t_{\max}} + 1\right) \quad (3)$$

式中: t 为当前迭代的次数; t_{\max} 为最大迭代次数。

权重系数 W 模拟了黏菌在遇到不同浓度的食物时生物振荡器的振荡频率的变化, 即

$$\overline{W}[R_1(i)] = \begin{cases} 1 + r \lg\left[\frac{F_b - S(i)}{F_b - F_w} + 1\right] & \text{条件 1} \\ 1 - r \lg\left[\frac{F_b - S(i)}{F_b - F_w} + 1\right] & \text{其他} \end{cases} \quad (4)$$

$$R_1 = \operatorname{sort}(S) \quad (5)$$

式中: 条件 1 表示适应度值 $S(i)$ 排名前一半的个体; $\operatorname{sort}(\cdot)$ 为排序函数; S 为 $S(i)$ 的集合; F_b 表示当前迭代过程中获得的最佳适应度值; F_w 表示当前迭代过程中获得的最差适应度值; R_1 表示适应度值从小到大的排序序列权重; W 向量主要模拟了黏菌静脉宽度与食物浓度之间的正负反馈机制, 当食物浓度高时 W 的值增大, 当食物浓度低时则值减小。

在利用 SMA 进行行星轮系设计模型优化求解时, 发现 SMA 常会出现无法根据算法搜索的实际情况来调整搜索行为、易倾向于原点和个体之间缺乏交互性等问题。

(1) SMA 使用参数 p 进行形态变换, 但是该方法存在缺陷。具体来说, 参数 p 通过 \tanh 函数对 SMA 的适应度进行计算, 在输入为正数时的有效定义域仅为 $[0, 3]$, 因此个体适应度差距较大时将超出 \tanh 函数的有效范围, 参数 p 将始终趋近于 1, 导致 SMA 搜索策略受到限制。此外, 在迭代后期, SMA 个体之间的适应度相近, 此时参数 p 也将始终趋近于 0, 导致 SMA 总是选择收缩的更新公式 $\vec{v}_c X(t)$, 使得 SMA 不能有效摆脱当前的搜索范围, 容易陷入局部最优值。

(2) SMA 使用参数 v_c 作为振荡参数, v_c 在 $[-1, 1]$ 随机取值, 其绝对值 a 从 1 衰减至 0, 所以存在一个向原点收缩的隐藏趋势。此外, 振荡参数 v_c 取值只能为 $[-1, 1]$, 因此其搜索范围也会受限且无法超出这个范围。

(3) SMA 在算法振荡过程中选用当前最优适应度个体 $X_b(t)$ 以及 2 个随机个体 $X_A(t)$ 、 $X_B(t)$ 作为位置更新的基础, 不能充分保证种群的多样性, 且迭代后期振荡作用较弱, 收缩机制不强, 导致算法容易陷入局部最优, 收敛速度慢。

针对 SMA 的以上 3 个缺陷, 本文提出的 WASMA 能够在 SMA 的基础上更好地根据当前搜索状况进行形态变换, 且增加了种群多样性, 进而从整体上增强了算法的探索能力和开发能力, 提升了算法的性能。

2.2 动态引导策略

在 SMA 数学模型中, 位置更新公式包含重分布 (finding food behavior)、振荡 (approaching food behavior) 和收缩 (wrapping food behavior) 3 种模式。SMA 使用参数 p 进行形态变换, 模拟黏菌觅食时的生理过程。该方法的不足之处在于, 通过适应度来计算算法是否需要模式转换的标准过于单一, 并且计算 p 时采用的双曲正切函数也存在一定缺陷。SMA 数学模型中参数 p 由式 (2) 计算。其中 \tanh 函数的有效定义域为 $[-3, 3]$, 如图 1 所示, 如果黏菌个体与当前最佳适应度的差值超过 3, 那么 p 就会直接趋向于 1, 导致 SMA 只会进行式 (1) 的第 2 行计算。

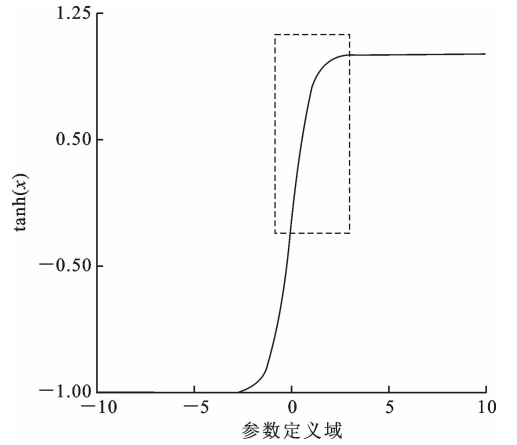
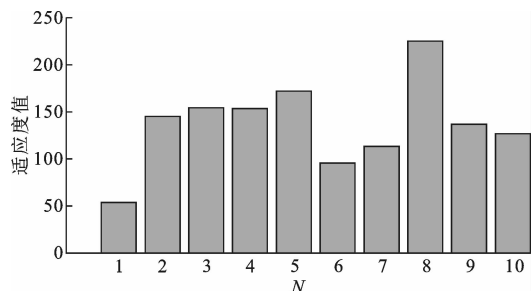


图 1 参数 p 的有效范围

Fig. 1 Effective range of parameter p

在解决某些问题时, SMA 在解空间搜索的过程中个体适应度差值始终保持在上述范围内, 原有的控制参数 p 便能够较好地引导种群进化; 在另一些函数的寻优过程中, 个体适应度的分布范围较大, 并且适应度值会随着位置变化而显著变化, 这时便会频繁出现上述情况, 即 p 始终维持在 1 左右, 导致 SMA 搜索策略单一, 如图 2 所示。

图 2 为算法迭代早期的个体适应度值分布及 p 大小, 描述了测试函数 Rastrigin 通过 SMA 在维度 $D=10$ 、种群数量 $N=10$ (黏菌个体编号 Agent1 ~ Agent10) 的条件下进行寻优, 种群个体在 30 次迭代时的适应度分布情况, 函数 $F(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i + 10)]$, 定义域为 $[-5.12, 5.12]$ 。可以

图2 函数迭代早期的个体适应度值分布及 p Fig. 2 Distribution of individual fitness values and p -values size in early stages of function iteration

看出,黏菌个体 $N=8$ 有着最高的适应度 225,并且与适应度次高的个体 $N=5$ 适应度差距超过 50;对于所有其他个体来说,与最优个体的适应度差值均超过了 $\tanh(\cdot)$ 的有效范围, p 值总是趋于 1。除此之外,到了 SMA 后期黏菌个体的适应度非常接近,导致 p 值直接趋近于 0,此时 SMA 只会进行式(1)的第 3 行计算,也会导致 SMA 的形态变换策略失效而局限于特定的搜索模式,使种群进化陷入停滞。

因此,本文引入最优解维持不变的进化代数 g 来衡量种群的进化状态,并以式(6)计算种群更改策略的概率以替换原有的 p 值计算方式,使种群在进化过程中处于停滞或者陷入局部最优解时,可以根据种群进化情况更好地选择进化策略,即

$$p = p_{\min} + (p_{\max} - p_{\min}) \frac{1}{1 + \exp\left[A\left(2\frac{g}{G} - 1\right)\right]} \quad (6)$$

式中: G 为最大停滞进化代数; p_{\max} 、 p_{\min} 、 A 均为控制参数,在试验中分别取 0.7、0.3 和 6。

改进后的参数 p 能够更加合理地控制种群的进化策略。 p 在进化停滞前期缓慢增长,留出一定的时间使种群依靠现有进化策略摆脱停滞。随着停滞代数的增加,算法确定种群陷入停滞状态。与原有参数在有效范围内接近线性的控制概率相比,WASMA 会更加显著地提升 p ,寻找新的策略,避免种群在现有的进化策略上浪费时间。为保持种群探索与算法收敛速度的平衡,当停滞代数不变时, p 值随着进化代数进一步上升的趋势就会缓和,进而提高种群跳出局部最优的可能性。

2.3 优化振荡过程

在 SMA 的振荡过程中,控制参数 v_c 对于 SMA 的寻优起到了十分重要的作用。在算法的迭代过程中, v_c 的绝对值从 1 递减到 0。这样的设计可以在迭代早期使种群广泛地散布于解空间当中,保持较

大的搜索范围并探索更多的区域;同时,随着迭代的进行,算法逐渐缩小搜索范围并进行细致的局部搜索,有助于最终收敛于最优解。

然而, v_c 的设计也为 SMA 带来了一定的偏差^[14]。具体来看, v_c 具有易趋向于原点的特性且有对称的探索空间,因此,当测试函数在原点或对称的解空间有全局最优解时,SMA 的收敛速度会变得更块,如图 3 所示。从图 3 可以看出,收缩的解并没有超出原有的界限,只向一个地方收缩。

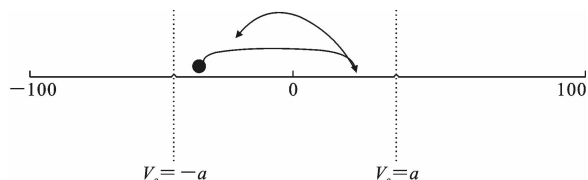


图3 SMA 的收缩过程

Fig. 3 Shrinkage process of SMA

综上所述,振荡参数 v_c 有 2 个缺陷,即易倾向于原点和搜索范围被局限(没有办法朝着图 3 中 2 条虚线之外的地方搜索)。针对此缺陷,本文提出一种改进的收缩过程,即

$$\vec{X}^* = \vec{X}(t) - v_c \cdot \text{sign}[\vec{X}(t) - \vec{X}(t-1)] \quad (7)$$

$$\text{sign} = f[\vec{X}(t)] - f[\vec{X}(t-1)] \quad (8)$$

式中: \vec{v}_c 是 $[0, 2]$ 的随机数; $\vec{X}(t)$ 是个体在 t 代的位置; \vec{X}^* 是个体改进的位置。

式(7)、式(8)以个体前一代和当代位置做差分进行位置更新,如此就可以在选择更好方向移动的同时,保留 SMA 以自身位置引导自身运动的特性,从而解决了其倾向于原点的缺陷。 $v_c \in [0, 2]$,且由 2 衰减至 0。当 $v_c > 1$ 时,SMA 就会朝着图 3 中 2 条虚线之外的地方搜索;当 $v_c < 1$ 时,算法就会在图 3 中 2 条虚线内搜索,从而解决了范围被限制的缺陷。

2.4 加权聚合学习机制

加权聚合思想源于人类的感知系统,在人脑接触到信息时,就会有意或者无意地对信息进行筛选,将注意力集中在某个特定的区域或对象上,以便更加有效地识别和处理复杂的信息。换句话说,人脑利用注意力机制为所获取的外部信息附加不同的权重,权重大的处理深度较深,权重小的处理深度较浅。本文将这一思想融入到群体智能算法中,并提出一种新的加权聚合学习机制,其基本思想为:在群体智能算法中,最优解的位置信息即为有效信息,其中包含不同维度的向量。而在每一代的不同个体中,都包含部分最优解的位置信息。因此本文设计

了一种类似于人脑注意力机制的方法,通过引入权重系数,依照种群的适应度倾向性地将不同个体之间的有效信息进行融合,进而产生新的聚合个体;同时,用新的聚合个体按照一定规则替换原先种群中的个体,从而维持整体种群数量不变。加权聚合学习机制的具体实现方法如下。

2.4.1 计算加权个体的数目

$$A_N = \text{round} \left[N_{\min} + (N_{\max} - N_{\min}) \frac{i_{t_{\max}} - i_t}{i_{t_{\max}}} \right] \quad (9)$$

式中: A_N 表示参与权值计算的个体数量; N_{\max} 和 N_{\min} 分别为最大加权个体数量和最小加权个体数量,在试验中分别取 N 和 5; i_t 和 $i_{t_{\max}}$ 分别代表当前迭代次数和最大迭代次数; $\text{round}(\cdot)$ 为取整函数。

在迭代初期,参与权值计算的个体与种群数量相近。这不仅可以加强收敛,还保证了种群多样性。到了迭代后期,种群中加权个体数量逐渐减少,降低了新个体对收敛的影响。

2.4.2 将全部个体按照适应度进行排序,选出前

A_N 个加权个体 A_X

$$A_X = X[\text{sort}(A_N)] \quad (10)$$

式中: A_X 为适应度排名靠前的 A_N 个加权个体; X 为全部个体; sort 为对 X 按照适应度排序后取前 A_N 个个体,算法总是会关注适应度较高的前 A_N 个个体。

2.4.3 对 A_X 的适应度进行量纲一化处理

$$F(X_i) = \frac{F(X_i) - F(X_{\min})}{F(X_{\max}) - F(X_{\min})} \quad (11)$$

式中: $F(X_i)$ 为 A_X 中第 i 个个体的适应度值; $F(X_{\max})$ 、 $F(X_{\min})$ 为 A_X 中最大适应度个体和最小适应度个体的适应度值。

经过量纲一化处理,个体适应度值被限定在 $[0, 1]$,以便下一步进行权重系数的计算。

2.4.4 计算权重系数

$$A_i = e^{F(X_i)} / \sum_{j=1}^{A_N} e^{F(X_j)} \quad (12)$$

式中: A_i 为第 i 个个体的权重系数。

式(12)可以使所有个体的权重集中在 $[0, 1]$,同时权重系数总和为 1,避免出现新产生的个体越界现象。

2.4.5 生成新的聚合个体

$$S' = \sum_{i=1}^N A_i x_i \quad (13)$$

式中: S' 为新生成的聚合个体; x_i 是第 i 个个体的位

置向量。

通过这样的方式,利用权重系数进行聚合学习产生全新的个体,该个体包含了从算法所关注的不同个体中提取出来的有效信息,可视化图像如图 4 所示。

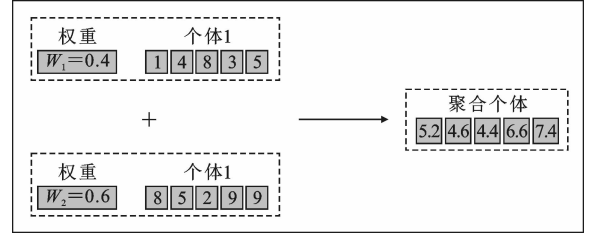


图 4 不同权重个体产生新的聚合个体

Fig. 4 New aggregated individuals generated by individuals with different weights

当生成新的聚合个体以后,根据其位置是否越界进行判断并做出修正,将全部个体 N 适应度重新排序,采用聚合个体替换适应度最接近的个体,以保证种群数量不变。

通过加权聚合学习机制的引入,不同个体的有效信息被集中、融合并产生新的个体,保证了种群的多样性,避免算法陷入局部最优,并实现了引导个体进化方向的作用。

2.5 WASMA 的复杂度分析

SMA 总体复杂度为 $O(D + TN(1 + \lg N + D))$,其中 $O(\cdot)$ 为复杂度函数; T 表示最大迭代次数。种群初始化的复杂度为 $O(D)$,适应度评估和排序的复杂度为 $O(N + N \lg N)$,权重更新的复杂度为 $O(ND)$,位置更新的复杂度为 $O(ND)$ 。

WASMA 在初始化时的复杂度为 $O(ND)$,适应度计算的复杂度为 $O(N)$,适应度排序的复杂度为 $O(N \lg N)$,权重更新的复杂度为 $O(ND)$,位置更新的时间复杂度为 $O(ND)$,加权聚合学习机制的复杂度为 $O(1)$ 。WASMA 总时间复杂度为 $O(ND + T(N(1 + \lg N + 2D) + 1))$,比 SMA 略微增加但是整体性能相近。

3 WASMA 在行星轮系优化设计中的性能验证

3.1 测试函数和对比较算法

对 WASMA 和 9 个对比较算法在 IEEE CEC2017^[15] 函数测试集和行星轮系设计试验数据上进行分析。9 个对比较算法分别为: SMA^[1]; AOSMA^[2]; BT β SMA^[16]; CNMSMA^[3]; DASMA^[5]; DFSMA^[6]; ELSHADE-SPACMA^[17]; AGSK^[18]; EBLSHADE^[19]。

3.2 试验环境及参数

SMA 具有良好的初始性能和较快的收敛速度, 所以将最大迭代次数设置为 200。在每组试验中, 算法独立运行 51 次, 对有种群的算法, 种群数量 $N=50$, 问题维度为 50。记录结果的最优值(F_{best})、平均值(F_{mean})、标准差(S_{id})进行对比和分析, 当标准差为 0 时说明试验数据没有波动, 算法的寻优能力处于完全稳定状态, 算法参数见表 1。

3.3 仿真试验和结果分析

为了验证 WASMA 的改进效果, 在相同环境下对 WASMA 和 9 个对比算法在函数测试集和行星轮系设计试验数据集上进行测试, 并通过数据对改进算法的有效性进行分析, 详见表 2~表 5。

表 1 相关对比算法参数

Tab. 1 Related comparison algorithm parameters

算法	参数设置
WASMA	$A=6, p_{\max}=0.7, p_{\min}=0.3, z=0.03$
SMA	$z=0.03$
AOSMA	$z=0.03$
BT β SMA	$z=0.03, K=30, \beta \in [0.1, 1], p=0.5$
CNMSMA	$z=0.03$, 混沌映射点 $x=0.7$
DASMA	$z=0.03, D_N=2$
DFSAM	$z=0.03, D_R=[0, 0.4]$
ELSHADE-SPACMA	$r^{init}=18D, p^{init}=0.3, p^{min}=0.15, H=5,$ $F_{cp}=0.5, c=0.8$
AGSK	$r^{init}=20D, p=0.05,$ $K_{WP}=[0.85, 0.05, 0.05, 0.05], c=0.05$
EBLSHADE	$r^{init}=18D, p=0.11, H=5, F_{cp}=0.5, c=0.8$

表 2 CEC2017 数据集中 10 个算法在 50 维下的试验结果(单峰函数)

Tab. 2 Experimental results of 10 algorithms in CEC2017 dataset at 50 dimensions (unimodal function)

函数	试验参数	WASMA	SMA	AOSMA	BT β SMA	CNMSMA	DASMA	DFSMA	ELSHADE-SPACMA	AGSK	EBLSHADE
F_1	最优值/ 10^8	1.440	103.000	0.174	713.000	119.000	46.800	0.221	57.800	176.000	349.000
	平均值/ 10^8	3.830	146.000	0.477	848.000	222.000	112.000	0.442	134.000	285.000	552.000
	标准差/ 10^8	2.990	25.400	0.171	61.500	40.800	37.400	0.135	35.000	39.000	92.500
F_3	最优值/ 10^5	1.020	1.240	1.400	1.400	1.100	1.360	1.410	1.140	2.720	0.437
	平均值/ 10^5	1.600	1.640	2.720	1.780	1.480	1.850	3.080	1.590	3.620	0.783
	标准差/ 10^4	3.470	3.030	7.360	1.940	1.640	3.000	7.770	2.440	3.890	2.510

注: 数字加粗为最优值, 下同。

表 3 CEC2017 数据集中 10 个算法在 50 维下的试验结果(多峰函数)

Tab. 3 Experimental results of 10 algorithms in CEC2017 dataset at 50 dimensions (multimodal function)

函数	试验参数	WASMA	SMA	AOSMA	BT β SMA	CNMSMA	DASMA	DFSMA	ELSHADE-SPACMA	AGSK	EBLSHADE
F_4	最优值/ 10^2	5.96	13.5	5.45	149.00	12.50	10.30	5.43	15.90	30.90	39.10
	平均值/ 10^2	7.76	20.00	6.55	195.00	26.10	17.20	6.46	24.80	48.90	102.00
	标准差/ 10^2	2.01	4.17	0.57	24.80	7.80	5.13	0.69	4.81	8.43	32.80
F_5	最优值/ 10^2	7.18	8.42	7.09	10.70	9.55	8.86	7.26	9.91	9.77	8.89
	平均值/ 10^2	7.94	9.26	8.25	11.50	10.20	9.61	8.27	10.70	11.20	9.60
	标准差/ 10^2	0.81	0.39	0.56	0.24	0.31	0.37	0.54	0.27	0.47	0.29
F_6	最优值/ 10^2	6.21	6.44	6.20	6.84	6.55	6.58	6.17	6.38	6.79	6.50
	平均值/ 10^2	6.33	6.60	6.39	6.97	6.76	6.71	6.43	6.50	6.98	6.76
	标准差	6.84	5.58	8.91	4.83	7.66	6.93	12.00	8.68	7.26	10.70
F_7	最优值/ 10^2	10.50	12.60	10.30	17.20	14.00	13.10	10.20	12.10	13.10	14.60
	平均值/ 10^2	12.30	13.90	11.50	18.70	15.40	15.30	11.80	13.00	14.50	17.60
	标准差/ 10^1	24.30	6.72	7.33	6.06	7.45	11.20	7.74	3.90	6.41	11.20
F_8	最优值/ 10^3	0.99	1.18	1.00	1.41	1.26	1.19	0.99	1.29	1.27	1.24
	平均值/ 10^3	1.10	1.25	1.11	1.47	1.34	1.28	1.09	1.38	1.41	1.32
	标准差/ 10^1	5.61	3.81	4.98	2.72	4.32	4.15	4.84	4.21	5.68	4.18
F_9	最优值/ 10^3	6.35	11.90	9.82	25.00	22.80	18.10	11.90	7.01	29.50	16.00
	平均值/ 10^4	1.44	2.13	1.92	3.52	3.07	2.54	2.02	2.54	4.65	3.35
	标准差/ 10^3	4.27	5.11	6.10	3.81	4.25	3.75	5.82	10.10	8.92	5.37
F_{10}	最优值/ 10^3	5.52	9.56	6.68	12.60	11.30	8.50	5.97	13.20	13.50	13.00
	平均值/ 10^3	8.32	11.60	8.26	14.50	13.30	10.30	8.37	14.80	14.50	14.60
	标准差/ 10^2	9.54	10.50	9.96	7.58	10.80	10.50	9.76	5.26	5.26	6.47

表 4 CEC2017 数据集中 10 个算法在 50 维下的试验结果(混合函数)

Tab. 4 Experimental results of 10 algorithms in CEC2017 dataset at 50 dimensions (mixed function)

函数	试验参数	WASMA	SMA	AOSMA	BT β SMA	CNMSMA	DASMA	DFSMA	ELSHADE-SPACMA	AGSK	EBLSHADE
F_{11}	最优值/ 10^3	1.32	1.80	1.36	13.50	3.84	2.97	1.34	3.45	4.73	2.21
	平均值/ 10^3	1.91	2.98	1.87	17.40	6.39	5.31	1.89	5.45	7.71	5.68
	标准差/ 10^2	5.22	7.06	7.86	20.20	15.10	14.80	7.94	10.80	24.70	19.00
F_{12}	最优值/ 10^7	0.84	19.00	0.68	3 060.00	128.00	38.80	1.24	56.30	217.00	58.70
	平均值/ 10^8	1.15	7.65	0.59	482.00	37.10	16.10	0.59	14.70	44.40	58.90
	标准差/ 10^8	3.84	4.35	0.37	87.10	12.80	8.70	0.32	5.00	14.30	52.80
F_{13}	最优值/ 10^5	0.56	21.10	0.59	63 200.00	910.00	134.00	0.51	448.00	959.00	7.86
	平均值/ 10^6	3.32	18.90	0.21	17 300.00	535.00	136.00	0.20	195.00	402.00	40.00
	标准差/ 10^6	21.70	17.50	0.12	5 220.00	301.00	118.00	0.13	107.00	256.00	112.00
F_{14}	最优值/ 10^5	0.55	0.64	2.67	7.41	2.81	6.34	1.16	0.46	1.36	0.02
	平均值/ 10^6	1.58	2.09	1.41	16.80	2.48	3.83	1.30	0.54	1.35	0.01
	标准差/ 10^6	1.98	2.91	0.96	14.00	1.84	3.19	1.03	0.41	0.92	0.01
F_{15}	最优值/ 10^5	0.07	0.38	0.20	936.00	24.90	3.42	0.14	20.80	8.41	0.13
	平均值/ 10^5	0.25	8.21	0.88	11 700.00	270.00	90.50	0.80	73.20	77.10	1.01
	标准差/ 10^5	0.13	15.10	0.69	4 720.00	240.00	143.00	0.66	61.80	92.90	0.97
F_{16}	最优值/ 10^3	2.59	2.37	2.77	4.93	3.54	2.97	2.72	4.87	5.27	3.69
	平均值/ 10^3	3.51	3.99	3.73	6.53	4.70	4.23	3.76	5.61	6.29	4.91
	标准差/ 10^2	4.70	4.75	4.87	6.99	5.36	5.91	4.38	2.40	4.35	4.22
F_{17}	最优值/ 10^3	2.37	2.52	2.98	4.03	2.85	2.85	2.74	3.87	4.17	2.64
	平均值/ 10^3	3.19	3.17	3.56	5.05	3.77	3.66	3.49	4.35	4.92	3.47
	标准差/ 10^2	3.40	2.99	3.51	6.05	3.81	3.80	3.70	2.46	3.32	3.98
F_{18}	最优值/ 10^6	0.34	0.63	0.90	9.49	3.05	1.74	0.74	0.99	2.57	0.02
	平均值/ 10^6	4.76	8.65	7.25	46.10	16.60	17.80	8.98	4.02	21.80	0.11
	标准差/ 10^6	4.27	8.61	6.89	26.50	14.20	16.90	5.92	2.64	10.80	0.08
F_{19}	最优值/ 10^6	0.01	0.12	0.01	137.00	1.94	0.75	0.01	1.60	0.41	0.03
	平均值/ 10^6	0.06	0.36	0.03	613.00	18.30	5.61	0.03	7.18	4.91	1.15
	标准差/ 10^6	0.23	0.15	0.02	459.00	13.90	3.90	0.02	4.59	4.77	2.29
F_{20}	最优值/ 10^3	2.45	2.61	2.70	3.06	2.75	2.98	2.61	3.14	3.59	2.88
	平均值/ 10^3	3.15	3.30	3.34	3.73	3.38	3.50	3.46	3.73	4.24	3.66
	标准差/ 10^2	3.14	2.80	3.35	2.62	2.27	2.80	2.95	1.51	2.04	2.25

测试函数集主要包含 29 个测试函数。其中, F_1 和 F_3 是单峰函数,没有局部最优,只有全局最优,这些函数主要用于检测算法的收敛性能; $F_4 \sim F_{10}$ 是具有局部极小值点的多峰函数,主要用于测试算法逃逸局部最优的能力; $F_{11} \sim F_{20}$ 是经过旋转或位移的混合函数,每个都包含 3 个或以上 CEC2017 基准函数,并赋予每个子函数一定的权重; $F_{21} \sim F_{30}$ 是由至少 3 个混合函数或旋转移位 CEC2017 基准函数组成的复合函数,每个子函数都有额外的偏移值和权重,进一步增大了优化难度。此外,在测试函数集中,原函数集中的 F_2 因为稳定性问题已经被官方移除。

从全部 10 个算法来看,WASMA 在 29 个测试函数下的表现综合性能最好,在收敛精度方面基本

位列前三,稳定性也存在较大优势。AOSMA、DFSMA 与 ELSHADESPACMA 等 3 个算法性能排名靠前。对比 SMA,改进后的 WASMA 除了在 F_{16} 多次运行结果下的最优值和 F_{17} 的平均值略低以外,在其余条件下的收敛精度均超过了 SMA。虽然 SMA 仅在 F_{16} 中取得最优解,但是在标准差上存在一定优势,SMA 的稳定性超过除 WASMA 以外的大多数算法。具体来看,单峰函数运算中 WASMA 的寻优能力相较于其他对比算法存在一定优势,这表明 WASMA 有较强的全局搜索能力。在处理多峰函数时,每个函数都存在 2 个或 2 个以上的局部极值,而 WASMA 在 $F_4 \sim F_8$ 中取得了前三的结果,在 F_9 和 F_{10} 得到了全算法的最优值,与 AOSMA 性能相似,都具有显著优势。这表明 WASMA

表 5 CEC2017 数据集中 10 个算法在 50 维下的试验结果(复合函数)
Tab. 5 Experimental results of 10 algorithms in the CEC2017 dataset at 50 dimensions (composite function)

函数	试验参数	WASMA	SMA	AOSMA	BTβSMA	CNMSMA	DASMA	DFSMA	ELSHADE-SPACMA	AGSK	EBLSHADE
F_{21}	最优值/ 10^3	2. 47	2. 63	2. 51	2. 94	2. 77	2. 62	2. 49	2. 73	2. 79	2. 72
	平均值/ 10^3	2. 56	2. 74	2. 59	3. 07	2. 85	2. 77	2. 60	2. 80	2. 92	2. 82
	标准差/ 10^1	4. 45	4. 73	4. 80	4. 88	4. 43	5. 06	6. 99	2. 68	5. 63	5. 16
F_{22}	最优值/ 10^3	2. 49	11. 10	7. 77	15. 50	4. 22	3. 41	7. 21	3. 76	14. 80	6. 01
	平均值/ 10^3	10. 40	13. 90	9. 92	16. 90	14. 40	13. 10	9. 97	6. 64	16. 30	12. 20
	标准差/ 10^3	1. 34	1. 07	0. 93	0. 48	2. 91	2. 64	1. 06	2. 62	0. 54	3. 36
F_{23}	最优值/ 10^3	2. 99	3. 17	2. 94	3. 94	3. 26	3. 16	2. 94	3. 19	3. 29	3. 24
	平均值/ 10^3	3. 15	3. 31	3. 05	4. 18	3. 38	3. 33	3. 04	3. 29	3. 53	3. 50
	标准差/ 10^1	9. 05	5. 95	7. 59	12. 40	7. 89	7. 95	6. 24	3. 64	9. 03	27. 50
F_{24}	最优值/ 10^3	3. 20	3. 45	3. 06	4. 23	3. 33	3. 29	3. 06	3. 36	3. 52	3. 38
	平均值/ 10^3	3. 43	3. 56	3. 21	4. 71	3. 48	3. 47	3. 22	3. 42	3. 72	3. 69
	标准差/ 10^2	1. 91	0. 68	1. 02	2. 05	0. 67	0. 89	0. 90	0. 28	1. 13	2. 19
F_{25}	最优值/ 10^3	3. 16	3. 60	3. 04	8. 65	3. 80	3. 53	3. 06	3. 82	4. 72	5. 30
	平均值/ 10^3	3. 32	3. 94	3. 12	10. 60	4. 59	3. 94	3. 13	4. 33	5. 79	7. 69
	标准差/ 10^2	2. 26	2. 10	0. 37	8. 70	3. 22	3. 14	0. 39	3. 29	5. 38	11. 20
F_{26}	最优值/ 10^3	3. 33	6. 05	3. 34	12. 70	6. 07	5. 44	3. 16	5. 55	9. 93	7. 82
	平均值/ 10^3	4. 75	9. 58	6. 64	14. 80	8. 23	7. 54	6. 97	7. 33	11. 30	11. 50
	标准差/ 10^3	1. 77	1. 08	1. 14	1. 05	1. 87	1. 85	1. 08	0. 89	0. 68	1. 62
F_{27}	最优值/ 10^3	3. 56	3. 74	3. 37	5. 61	3. 60	3. 64	3. 32	3. 81	4. 21	3. 54
	平均值/ 10^3	3. 79	4. 02	3. 59	6. 43	3. 84	3. 92	3. 55	4. 03	4. 56	3. 85
	标准差/ 10^2	1. 28	1. 52	0. 96	3. 65	1. 46	1. 88	0. 97	1. 27	1. 71	1. 90
F_{28}	最优值/ 10^3	3. 48	4. 23	3. 32	8. 77	4. 51	3. 91	3. 36	4. 49	5. 30	5. 44
	平均值/ 10^3	3. 64	4. 67	4. 25	9. 90	5. 34	4. 65	3. 71	5. 10	6. 28	7. 18
	标准差/ 10^2	0. 88	1. 99	13. 40	6. 55	4. 15	3. 63	3. 56	3. 04	3. 76	8. 71
F_{29}	最优值/ 10^3	3. 68	4. 78	4. 18	8. 79	5. 44	5. 16	4. 19	5. 13	6. 63	5. 12
	平均值/ 10^3	4. 66	5. 80	5. 07	13. 80	6. 46	6. 01	5. 10	6. 51	7. 87	6. 17
	标准差/ 10^2	3. 63	5. 28	4. 43	35. 60	6. 31	5. 32	4. 63	3. 59	6. 91	5. 57
F_{30}	最优值/ 10^7	0. 66	5. 82	0. 48	43. 00	13. 30	8. 05	0. 64	7. 84	9. 45	3. 01
	平均值/ 10^8	0. 17	1. 35	0. 19	11. 50	2. 82	1. 85	0. 19	1. 54	1. 49	0. 89
	标准差/ 10^8	0. 18	0. 48	0. 06	5. 02	1. 04	0. 72	0. 08	0. 42	0. 46	0. 44

具有很强的跳出局部最优的能力。在混合函数时, WASMA 表现极佳,在部分函数如 F_{11} 、 F_{15} 、 F_{17} 和 F_{20} 中取得了最高的收敛精度,在 F_{12} 、 F_{13} 、 F_{16} 、 F_{18} 中获得了第 2 的最优值,说明 WASMA 在处理复杂、多模态问题时的性能良好。复合函数通常具有高度非线性、非凸和多模态的特性,用于评估优化算法在处理高度复杂问题时的性能,以及对问题结构的理解和建模能力。在复合函数的测试中,WASMA 以较为明显的优势超过 AOSMA 和高性能算法 ELSHADESPACMA,且在 F_{20} 、 F_{21} 、 F_{22} 、 F_{29} 四个函数中都取得了最优解以及在 F_{20} 、 F_{21} 、 F_{26} 、 F_{28} 、 F_{29} 、 F_{30} 中都取得了最优平均值,证明了 WASMA 对于复杂问题仍旧具有较强的适应能力。

图 5 为 10 个算法在 CEC2017 数据集 50 维下的收敛曲线,按照单峰函数(F_1 、 F_3)、多峰函数(F_4 、 F_6 、 F_8)、混合函数(F_{11} 、 F_{15} 、 F_{18} 、 F_{19})以及复合函数(F_{21} 、 F_{25} 、 F_{28})分为 4 组。从收敛精度看,WASMA 在单峰函数 F_1 以外的 4 组函数中都具有显著优势。从收敛速度看,该算法在迭代前期(迭代次数小于 50)快速收敛,随后逐渐向最优解靠近。与 SMA 相比,本文算法取得了明显的改善,且与其余改进算法和高性能算法相比也有一定的优势。

从 WASMA 的改进策略进行分析,形态变换参数的改进使算法不只是依靠适应度来判别进化策略,这进一步提升了在不同函数下的收敛速度和稳定性;振荡参数 v_c 的改进解决了算法向原点靠近的

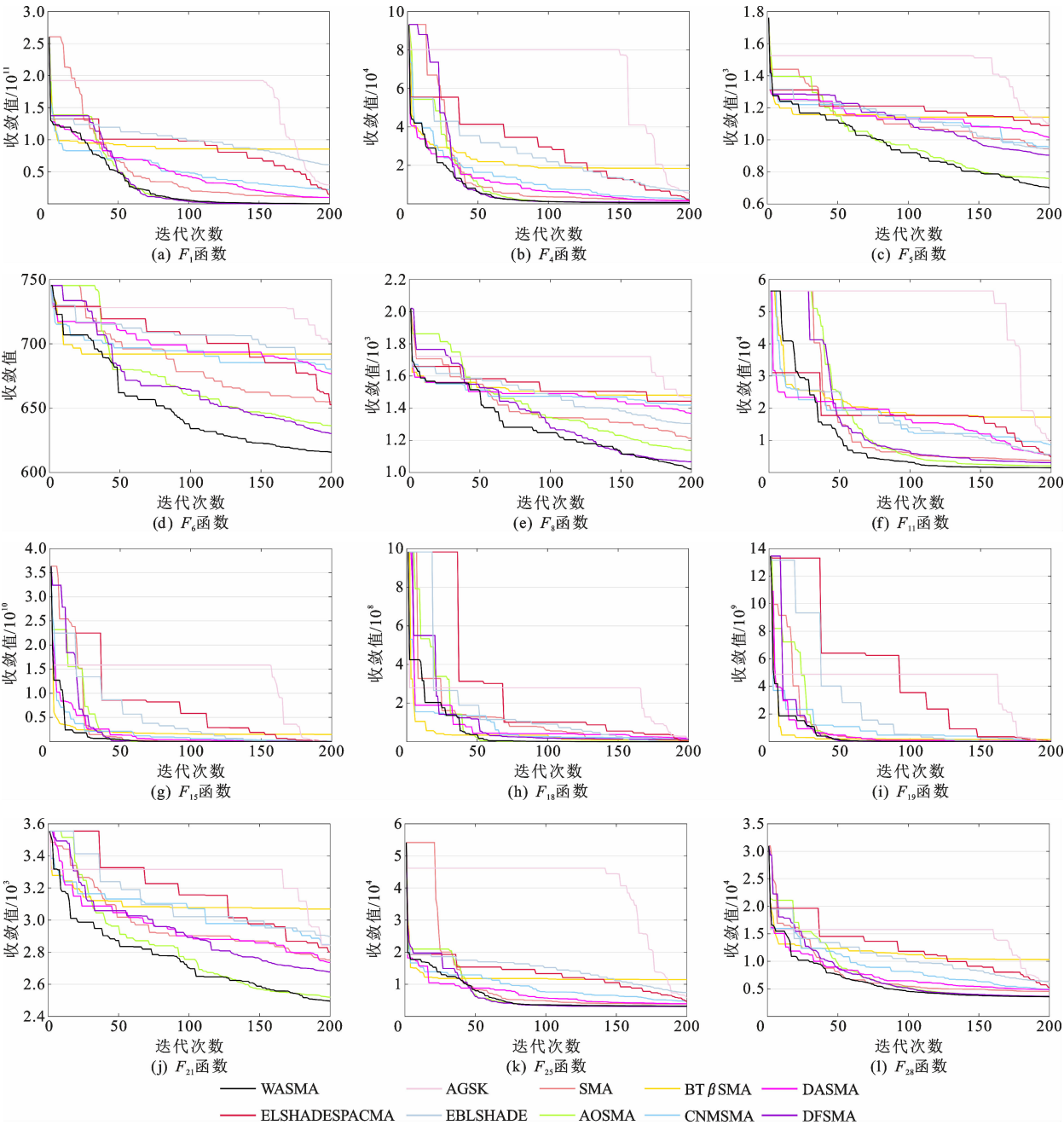


图5 CEC2017数据集中10个算法在50维下的收敛曲线

Fig. 5 Convergence curves of 10 algorithms in CEC2017 dataset at 50 dimensions

趋势,同时扩大了收缩过程的搜索范围,使算法在引入偏差的测试函数中具有更强的适应性,并且在多维函数下具有更强的跳出局部最优的能力。加权聚合学习机制的引入保证了种群多样性,更好地引导了种群的进化方向,提高了收敛速度和收敛精度。试验表明,3种改进策略的融合使得改进后的WASMA在整体性能上具有显著优势,证明了算法改进的有效性。

表6为WASMA和其他9种对比算法行星轮系设计问题优化结果。根据试验结果可以看出,

WASMA的综合优化结果在行星轮系设计问题中最少,寻优结果最好,验证了WASMA能够在处理函数优化问题中保持性能良好的同时,在复杂的工程优化问题中也有一定适用性。

4 结 语

(1)加权聚合学习机制通过有效整合不同个体的优秀信息,显著提高了算法的搜索效率和收敛速度。在行星轮系设计优化过程中,这种机制使得算法能够更快速地找到全局最优解,从而显著提升了

表 6 行星轮系设计问题优化结果
Tab. 6 Optimization results of planetary gear system design problems

算法	优化变量									优化结果
	N_1	N_2	N_3	N_4	N_5	N_6	P	m_1	m_2	
WASMA	34.80	25.50	24.80	24.20	15.60	87.30	1.26	1.87	1.97	0.53
SAM	72.60	24.30	13.50	30.40	21.60	110.00	2.27	0.98	0.51	0.56
AOSMA	17.60	13.60	17.90	16.80	13.50	62.10	1.25	4.17	0.52	0.55
BT β SMA	21.60	13.50	13.50	16.50	13.50	49.80	0.51	0.51	0.51	0.85
CNMSMA	30.50	17.70	14.30	16.80	18.30	61.70	0.62	0.53	0.72	0.54
DASMA	28.50	17.20	13.50	16.50	13.50	61.60	0.51	0.56	0.76	0.54
DFSMA	23.90	13.50	13.50	16.50	18.50	62.10	0.52	2.72	0.96	0.54
ELSHADE-SPACMA	24.30	13.50	13.50	16.90	15.00	61.50	0.96	3.13	0.51	0.54
AGSK	69.40	18.30	14.80	34.50	22.90	103.00	1.89	1.13	1.15	1.32
EBLAHADE	30.40	18.60	14.80	17.40	18.60	62.10	0.76	1.87	1.76	0.54

行星轮系的性能。

(2)在行星轮系设计优化中,需要同时考虑多个性能指标,如传动效率、体积、噪音等。本文提出的 WASMA 能够在多个目标之间取得更好的平衡,得到更加全面和满意的优化结果。通过引入加权聚合学习机制,优化行星轮系的传动比、齿轮齿数、模数等关键参数,可以显著提高行星轮系的传动效率,降低噪音和振动水平,同时满足行星轮系结构的体积和重量的限制要求。

(3)本文还探讨了 WASMA 参数对优化结果的影响,并给出了相应的参数调整建议,有助于进一步提高算法的优化性能。该算法不仅能够有效提升行星轮系的性能,而且能够处理多目标优化问题,为机械传动领域的相关研究提供了新的思路和方法。

参考文献:
References:

[1] LI S,CHEN H,WANG M,et al. Slime mould algorithm:A new method for stochastic optimization[J]. Future generation computer systems, 2020, 111: 300-323.

[2] NAIK M K,PANDA R,ABRAHAM A. Adaptive opposition slime mould algorithm[J]. Soft computing,2021,25(22):14297-14313.

[3] LIU Y,HEIDARI A A,YE X,et al. Boosting slime mould algorithm for parameter identification of photovoltaic models[J]. Energy,2021,234:121164.

[4] RIZK-ALLAH R M,HASSANIEN A E,SONG D. Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites [J]. ISA transactions, 2022, 121: 191-205.

[5] ZHAO S,WANG P,HEIDARI A A,et al. Multilevel threshold image segmentation with diffusion association slime mould algorithm and Renyi's entropy for chronic obstructive pulmonary disease[J]. Computers in Biology and Medicine,2021,134:104427.

[6] HU J,GUI W,HEIDARI A A,et al. Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection [J]. Knowledge-Based Systems, 2022,237:107761.

[7] 贾鹤鸣,刘宇翔,刘庆鑫,等. 融合随机反向学习的黏菌与算术混合优化算法[J]. 计算机科学与探索, 2022,16(5):1182-1192.

[8] JIA He-ming, LIU Yu-xiang, LIU Qing-xin, et al. Hybrid algorithm of slime mould algorithm and arithmetic optimization algorithm based on random opposition-based learning[J]. Journal of Frontiers of Computer Science and Technology, 2022, 16 (5): 1182-1192.

[9] ABUALIGAH L,DIABAT A,MIRJALILI S,et al. The arithmetic optimization algorithm[J]. Computer Methods in Applied Mechanics and Engineering, 2021,376:113609.

[10] 卢万杰,陈子林,付 华,等. 多策略融合的改进黏菌算法及其应用[J]. 智能系统学报,2023,18(5): 1060-1069.

[11] LU Wan-jie, CHEN Zi-lin, FU Hua, et al. Improved slime mould algorithm with multistrategy integration and its application[J]. CAAI Transactions on Intelligent Systems,2023,18(5):1060-1069.

化的无人机三维路径规划[J]. 上海交通大学学报, 2023, 57(10):1282-1291.

HUANG He,GAO Yong-bo,RU Feng,et al. 3D path planning of UAV based on adaptive slime mould algorithm optimization[J]. Journal of Shanghai Jiao Tong University,2023,57(10):1282-1291.

[11] 翟卫欣,潘家文,兰玉彬,等. 基于多元振荡黏菌算法的田路分割模型参数优化方法[J]. 农业工程学报, 2022,38(18):176-183.

ZHAI Wei-xin,PAN Jia-wen,LAN Yu-bin,et al. Parameter optimization of field-road trajectory segmentation model using multiplex oscillation slime mould algorithm[J]. Transactions of the Chinese Society of Agricultural Engineering,2022,38(18):176-183.

[12] 谢国民,林忠宝. 多策略改进黏菌算法阶段优化 HSVM 变压器故障辨识[J]. 电子测量与仪器学报, 2024,38(3):67-76.

XIE Guo-min,LIN Zhong-bao. ISMA algorithm stage optimization for HSVM transformer fault identification[J]. Journal of Electronic Measurement and Instrumentation,2024,38(3):67-76.

[13] ABDERAZEK H,SAIT S M,YILDIZ A R. Optimal design of planetary gear train for automotive transmissions using advanced meta-heuristics[J]. International Journal of Vehicle Design,2019,80(2/3/4):121-136.

[14] QIAN Q,DENG Y,SUN H,et al. Enhanced beetle antennae search algorithm for complex and unbiased optimization[J]. Soft Computing,2022,26(19):10331-10369.

[15] WU G,MALLIPEDDI R,SUGANTHAN P N. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization[R]. National University of Defense Technology, and Kyungpook National University, and Nanyang Technological University,2017.

[16] SUN K,JIA H,LI Y,et al. Hybrid improved slime mould algorithm with adaptive β hill climbing for numerical optimization[J]. Journal of Intelligent & Fuzzy Systems,2021,40(1):1667-1679.

[17] HADI A A,MOHAMED A W,JAMBI K M. Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm[J]. Heuristics for optimization and learning,2021(906):103-121.

[18] MOHAMED A W,HADI A A,MOHAMED A K,et al. Evaluating the performance of adaptive gaining-sharing knowledge based algorithm on CEC 2020 benchmark problems[C]//IEEE. Proceedings of 2020 IEEE congress on evolutionary computation (CEC). New York:IEEE,2020:1-8.

[19] MOHAMED A W,HADI A A,JAMBI K M. Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization[J]. Swarm and Evolutionary Computation,2019,50:100455.