

嵌入 Cat 映射的混合变异探路者算法及其应用

毛雪迪,王 冰,夏煌智,张鲁平,李永超
(牡丹江师范学院 数学科学学院,黑龙江 牡丹江 157000)

摘 要:针对探路者算法(PFA)求解精度不高、寻优速度较慢与易陷入局部最优等问题,提出一种嵌入 Cat 映射的混合变异探路者算法(CHMPFA)运用于函数优化问题中。首先,利用 Cat 混沌映射的随机性和分散性等特点,再结合反向学习的引导作用,使种群能够覆盖在更为广阔的搜索空间,提高算法的全局搜索能力;其次,在探路者位置更新阶段引入衰减因子平衡算法的全局和局部搜索能力,通过迭代次数的增长逐渐地缩小搜索空间范围,帮助算法快速找到最优解,从而提升算法的搜索效率和收敛速度;最后,利用变异概率随机选择柯西变异或高斯变异对最优个体进行位置扰动,两种变异策略能够帮助个体快速跳出局部最优向其它区域前进。将 CHMPFA 在 10 个经典基准测试函数和 12 个复杂的 CEC2017 函数集上进行测试,并将其运用于压力容器工程设计问题,与原算法和其它算法的实验结果进行比较,结果表明 CHMPFA 的求解精度、寻优速度与局部最优规避性均明显加强,更低的工程造价成本进一步验证了 CHMPFA 的鲁棒性。

关键词:探路者算法;函数优化问题;Cat 映射;混合变异;工程优化问题

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2024)02-0171-09

doi:10.3969/j.issn.1673-629X.2024.02.025

Hybrid Mutation Pathfinder Algorithm Embedded with Cat Mapping and Its Application

MAO Xue-di, WANG Bing, XIA Huang-zhi, ZHANG Lu-ping, LI Yong-chao
(School of Mathematical Sciences, Mudanjiang Normal University, Mudanjiang 157000, China)

Abstract: A hybrid mutation pathfinder algorithm embedded with Cat mapping (CHMPFA) is proposed for the function optimization problem in view of the problems of low accuracy of pathfinder algorithm (PFA) solution, slow speed of finding the best and easy to fall into local optimum. Firstly, using the characteristics of Cat chaotic mapping such as randomness and dispersion, combined with the guiding effect of opposition-based learning, the population can cover a wider search space and improve the global search capability of the algorithm. Secondly, the introduction of reduction factors in the pathfinder position update phase balances the global and local search capabilities of the algorithm, gradually narrowing the search space range through the growth of the number of iterations, helping the algorithm to find the optimal solution quickly, thus enhancing the search speed and convergence of the algorithm. In the end, the optimal individual is perturbed in position using the mutation probability of randomly selected Cauchy mutation or Gaussian mutation, and the two mutation strategies can help individual quickly jump out of the local optimum to other regions. The CHMPFA is tested on 10 classical benchmark test functions and 12 complex CEC2017 function, and applied to pressure vessel engineering design problem. The experimental results are compared with those of the original algorithm and other algorithms, and the results show that the CHMPFA is significantly enhanced in terms of solution accuracy, finding speed and local optimum avoidance, and the lower engineering cost further validate the robustness of the CHMPFA.

Key words: pathfinder algorithm; function optimization problem; Cat map; hybrid mutation; engineering optimization problem

0 引 言

智能优化算法是一类基于自然界生物进化、群体智能、神经网络等理论和方法而发展起来的优化算法,这类算法通过模拟生物进化、群体社会行为等机制寻

找最优解或次优解,具有全局优化、鲁棒性强等特点,在许多领域中得到了广泛应用。2019年,Yapici等人提出了一种新的群智能优化算法,即探路者算法(Pathfinder Algorithm, PFA)^[1],它基于生物进化理论

收稿日期:2023-05-10

修回日期:2023-09-12

基金项目:牡丹江师范学院科研项目(GP2020003);国家自然科学基金资助项目(12271036);牡丹江师范学院校级课题项目(kjcx2022-097mdjnu, kjcx2022-019mdjnu)

作者简介:毛雪迪(1997-),女,硕士研究生,研究方向为智能计算;通讯作者:王 冰(1978-),男,博士,教授,研究方向为智能计算。

和群体智能原理,通过模拟生物进化和自然选择的过程寻找最优解。该算法具有自适应参数设置、多种探索策略、鲁棒性强和高效性等特点,已经在多个领域得到了广泛的应用,如机器学习、信号处理、图像处理等,然而,PFA 也存在收敛速度慢、求解精度不高、容易陷入局部最优等问题。文献[2]通过借助 PFA 中探路者与跟随者独特的更新模式,成功地提升了灰狼优化算法的寻优性能,这种融合方式得到的优化算法具有较高的收敛精度。文献[3]在 PFA 中引入了一种向导机制,使得带有向导机制的探路者个体能够收集周围个体的优良信息,并将这些信息传播给其它个体,同时,还引入了一种新型的变异概率 pcR,以增强算法跳出局部最优的能力。文献[4]将教与学算法与 PFA 相结合,通过在跟随者阶段中引入指数步长算子优化跟随者个体提高了算法整体的寻优精度和收敛速度。文献[5]利用动态反向学习策略提升了初始种群质量,提出一种新的跃迁档案保存并创造新的最优个体,引导陷入局部最优的个体跳出当前位置;又提出一种双跳模型协调算法的全局搜索和局部开发能力。

为解决上述问题,该文提出一种嵌入 Cat 映射的混合变异探路者算法(CHMPFA)。首先,利用混沌反向学习策略增加了初始种群的多样性和分散性;其次,在探路者更新公式中加入衰减因子,在迭代过程中控制探路者的搜索步长从而控制搜索范围;然后,利用随机数对最优个体进行混合变异,帮助其脱离当前区域向全局最优值靠近;最后,通过在 10 个基准函数和 12 个 CEC2017 基准测试函数进行仿真实验验证策略的鲁棒性,并将 CHMPFA 应用于压力容器工程问题中,优秀的寻优结果进一步证明了改进算法的有效性。

1 标准探路者算法

在标准探路者算法中,种群只分为两种类型:探路者和领导者,探路者为适应度值最好的个体,其余个体为跟随者的角色。

探路者算法中的位置向量 X 将由 N 个维度为 d 的种群个体组成,每个个体定义为 $X_i = [x_{i1}, x_{i2}, \dots, x_{id}]$, 各维度的上界为 $ub = [ub_1, ub_2, \dots, ub_d]$, 下界为 $lb = [lb_1, lb_2, \dots, lb_d]$, 即种群随机初始化为:

$$X_{N \times d} = \text{rand}(N, d) \times (ub - lb) + lb \quad (1)$$

其中, N 为种群规模, d 为空间维度;在种群迭代过程中,选出适应度值最优的个体作为探路者,跟随者均向它移动,更新方式为式 2:

$$x_p^{t+1} = x_p^t + 2r_1(x_p^t - x_p^{t-1}) + A \quad (2)$$

$$A = u_1 \times \exp\left\{\frac{-2t}{T_{\max}}\right\} \quad (3)$$

其中, t 为当前迭代次数; T_{\max} 为算法的最大迭代次

数; x_p^t 表示在第 t 次迭代时的探路者位置; x_p^{t-1} 表示 $t-1$ 代探路者位置; x_p^{t+1} 表示 $t+1$ 代探路者位置; $r_1 \in [0, 1]$ 为探路者移动的步长因子; u_1 为 $[-1, 1]$ 范围内的随机数; A 为探路者移动的随机步长,由 u_1 的取值决定其多向性。

探路者更新完后,跟随者根据探路者位置进行更新,更新方式为式 4:

$$x_i^{t+1} = x_i^t + R_1(x_j^t - x_i^t) + R_2(x_p^t - x_i^t) + \varepsilon \quad (4)$$

$$\varepsilon = (1 - \frac{t}{T_{\max}})u_2D_{ij} \quad (5)$$

$$D_{ij} = \|x_i - x_j\| \quad (6)$$

$$R_1 = \alpha r_2 \quad (7)$$

$$R_2 = \beta r_3 \quad (8)$$

其中, x_i^t 与 x_j^t 分别表示第 i 个与第 j 个跟随者在第 t 次迭代时的位置; x_i^{t+1} 表示第 i 个跟随者更新后的位置; R_1 和 R_2 为随机向量; α 为跟随者与跟随者之间的相互作用系数, β 为探路者与跟随者之间的相互作用系数, α 和 β 均为取值在 $[1, 2]$ 之间的服从均匀分布的向量, r_2 和 r_3 均为 $[0, 1]$ 范围内服从均匀分布的随机数; ε 表示是为跟随者移动提供随机性的因子, u_2 为 $[-1, 1]$ 范围内的随机数,其决定着跟随者的移动方向, D_{ij} 为第 i 个跟随者与第 j 个跟随者之间的距离。

2 算法改进

2.1 混沌反向学习初始化种群

为了改善种群质量和提高算法的搜索性能,该文引入 Cat 混沌映射结合反向学习的方法^[6]初始化种群。Cat 映射是一个结构简单的二维可逆混沌映射,可以产生具有随机性、分散性和复杂性的种群分布,能够帮助算法从多个起点同时进行搜索,增加算法的全局搜索能力,其表达式如下:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1 \quad (9)$$

反向学习^[7]则是一种基于贪心策略的启发式算法,可以在有限的时间内快速生成优秀的种群,增加算法的局部搜索能力。将两种方法结合起来使用,能够在初始化种群时既考虑全局性又考虑局部性,提高算法的搜索效率和多样性。此外, Cat 混沌映射和反向学习还可以互相补充彼此的不足。Cat 混沌映射在产生种群时具有一定的随机性,容易产生多样性,但可能会受到局部最优解的影响,难以快速收敛;而反向学习虽然可以快速生成优秀的种群,但可能会陷入局部最优解,将两种方法结合使用可以平衡二者之间的优缺点,提高算法的搜索性能和鲁棒性。

由混沌序列结合反向学习初始化种群,其步骤是:首先利用 Cat 映射产生初始种群 $x_i (i = 1, 2, \dots, N)$, 然

后为每个初始解按公式 10 求其相应的反向解。

$$x_i^{\text{op}} = \text{rand} \times (\text{ub}_i^j - \text{lb}_i^j) - x_i \quad (10)$$

其中, x_i^{op} 为 x_i 对应的反向解, rand 是区间 $[0, 1]$ 之间的随机数, ub_i^j 和 lb_i^j 分别表示所有初始解中第 j 维上的上界与下界。最后, 初始解 x_i 和反向解 x_i^{op} 合并为新种群 $\{x_i \cup x_i^{\text{op}}\}$, 种群数量由 N 增至 $2N$, 计算新种群 $\{x_i \cup x_i^{\text{op}}\}$ 适应度值并进行排序, 选取适应度值排名前 N 的解作为最终初始解。

2.2 赋予衰减性的探路者更新方式

为解决探路者算法搜索效率和求解精度低等问题, 该文提出将衰减因子引入 PFA 探路者更新公式中, 随着迭代的进程不断降低衰减因子的大小, 算法搜索空间的范围也会逐渐缩小至最优解附近, 从而避免了算法过早陷入局部最优区域的状况, 这能够使算法快且准地找到理论最优解, 帮助算法改善了搜索效率和求解精度。提出的衰减因子是一个非线性递减函数, 定义式如下:

$$w = \exp(-st^{\frac{1}{10}}) \quad (11)$$

其中, $s \in [0, 1]$ 为衰减系数 (该文选取 $s = 0.5$), 控制算法收敛速度, t 为当前迭代次数。引入衰减因子后的探路者更新公式如下:

$$x_p^{t+1} = w(x_p^t + 2r_1(x_p^t - x_p^{t-1}) + A) \quad (12)$$

通过衰减因子的非线性变化, 使算法在不同阶段的搜索更加高效。在算法初期, 较大的衰减因子帮助探路者在解空间中进行更广泛的搜索; 随着算法的迭代, 较小的衰减因子帮助算法对局部区域进行更加精确的探索。

2.3 混合变异策略

(1) 柯西变异。

柯西变异来源于柯西分布^[8], 一维柯西概率密度公式如下:

$$f(x) = \frac{1}{\pi} \times \frac{\lambda}{\lambda^2 + x^2}, x \in (-\infty, +\infty) \quad (13)$$

当 $\lambda = 1$ 时, 是标准柯西分布。从图 1 的标准柯西、高斯分布概率密度函数曲线观察可知, 柯西分布与高斯分布类似, 主要差异表现为柯西分布峰值略小于高斯分布, 柯西分布在原点处有较高的概率密度, 在两端有较低的概率密度, 柯西分布的边缘较长且扁平, 比高斯分布更平缓的逼近于 0。

根据此特点对探路者和跟随者位置更新后的最优个体进行柯西变异, 通过柯西算子的扰动特性提升算法的寻优性能。当最优个体无法跳出局部最优解时, 柯西算子能够提供较长的步长帮助其个体跳出局部最优; 当最优个体向理论最优值靠近时, 柯西算子能够提供较短的步长提升个体收敛速度。柯西变异策略数学模型如下:

$$x_{\text{newbest}}^t = x_{\text{best}}^t + \text{cauchy}(0, 1) \times x_{\text{best}}^t \quad (14)$$

其中, x_{best}^t 为最优个体, x_{newbest}^t 为变异后最优个体, $\text{cauchy}(0, 1)$ 为标准柯西分布。

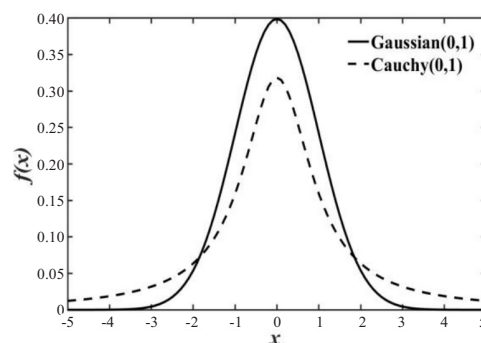


图1 标准柯西、高斯分布概率密度函数曲线

(2) 高斯变异。

高斯算子可以防止种群陷入局部最优, 增加种群的多样性。为了提升 PFA 局部搜索的能力和跳出局部最优的概率, 对最优个体进行高斯变异操作, 高斯概率密度公式和高斯变异策略数学模型如下:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (15)$$

$$x_{\text{newbest}}^t = x_{\text{best}}^t \times (1 + \text{Gaussian}(\sigma)) \quad (16)$$

其中, μ 表示期望, σ 表示标准差, 标准正态分布的期望和方差分别设置为 0 和 1, $\text{Gaussian}(\sigma)$ 是由式 15 形成的标准正态分布的随机变量。

由图 1 可知高斯分布的概率密度函数呈现钟形曲线, 均值即为期望值, 并且其方差较大、分布范围较广, 所以当最优个体处于局部最优解时, 高斯变异可以让其位置发生微小变化, 使得搜索过程可以继续向周围未被探索的区域扩展, 从而增加逃离局部最优的可能性。

(3) 混合变异。

在 PFA 中引入柯西变异和高斯变异可以使得算法更加全面地探索解空间, 从而有更大的可能找到最优解。柯西变异可以增加算法的全局搜索能力, 因为它具有长尾分布, 可以探索到远离当前搜索点的区域; 而高斯变异则可以增加算法的局部搜索能力, 因为它可以更加精准地探索周围区域, 从而更容易找到局部最优解。所以, 该文根据变异概率 p 将柯西变异和高斯变异融合为混合变异策略引入 PFA 当中, 如公式 17 所示:

$$x_{\text{newbest}}^t = \begin{cases} x_{\text{best}}^t + \text{cauchy}(0, 1) \times x_{\text{best}}^t & p < 0.5 \\ x_{\text{best}}^t \times (1 + \text{Gaussian}(\sigma)) & p \geq 0.5 \end{cases} \quad (17)$$

但是, 变异后最优个体位置无法保证优于原始位置, 所以在混合变异位置扰动后加入保优策略, 即贪婪策略, 通过比较变异后最优个体与原最优个体的适应

度值,选取适应度值最好的个体作为当前代的全局最优解,则把变异后个体替换掉原始个体,判断公式如下:

$$x_{\text{newbest}}^t = \begin{cases} x_{\text{best}}^t & f(x_{\text{best}}^t) < f(x_{\text{newbest}}^t) \\ x_{\text{newbest}}^t & f(x_{\text{best}}^t) \geq f(x_{\text{newbest}}^t) \end{cases} \quad (18)$$

2.4 CHMPFA 算法伪代码

CHMPFA 算法伪代码如下:

```
Initialize PFA parameters
Initialize population using chaoticopposition-based learning
Calculate the fitness of initial population
Find the pathfinder
While  $t < \text{maximum number of iterations}$ 
    Update pathfinder the position using Eq. (12) and check
    the bound
    if new pathfinder is better than old
        Update pathfinder
    end
    for  $i = 2$  to maximum number of populations
        Update follower the position using Eq. (4) and check
        the bound
    end
    Calculate new fitness of members
    Find the best fitness
    if  $\text{rand} < 0.5$ 
        Update best number position using Eq. (14)
    else
        Update best number position using Eq. (16)
    end
    Calculate the fitness of new best number
    if new best fitness < best fitness
        Update best member
    end
    Calculate new fitness of members
    Find the best fitness
    if best fitness < fitness of pathfinder
        Pathfinder = best member
        Fitness = best fitness
    end
     $t = t + 1$ ;
end While
```

2.5 时间复杂度分析

时间复杂度是体现算法运行效率的重要指标,假设种群规模为 N ,空间维度为 d ,最大迭代次数为 T_{\max} , $f(x)$ 表示适应度函数。根据标准 PFA 流程可知,PFA 的总时间复杂度为 $O(d + f(d))$,现对 CHMPFA 的时间复杂度进行分析。

对于 CHMPFA,在初始化种群阶段,假设初始化算法参数的时间为 t_1 ,按适应度值对所有个体进行排序并选定探路者的时间为 t_2 ,由于通过混沌反向学习

初始化种群个体的时间复杂度为 $O(N \times d)$,且对每个个体计算适应度值的时间为 $f(d)$,则初始化种群阶段的时间复杂度为:

$$T_1 = O(t_1 + N \times f(d) + t_2) + O(N \times d) = O(d + f(d))$$

进入迭代期,在探路者阶段,探路者数目为 1,假设计算自适应权重 w 的时间为 t_3 ,生成随机数 r_1 需要的时间为 t_4 ,计算 A 的时间为 t_5 ,更新探路者位置时间为 t_6 ,对探路者个体的每一维边界处理的时间为 t_7 ,则探路者阶段的时间复杂度为:

$$T_2 = O(t_3 + 1 \times (t_4 + t_5 + t_6 + t_7) \times d) = O(d)$$

在跟随者阶段,CHMPFA 更新跟随者位置的方式与标准 PFA 保持一致,所耗费的时间并没有增加。因此,跟随者阶段的时间复杂度也应与标准 PFA 一致为:

$$T_3 = O(d + f(d))$$

在变异阶段,最优个体进行柯西变异或高斯变异的时间为 t_{12} ,计算最优个体适应度值的时间为 $f(d)$,利用贪婪策略判断新最优个体是否代替旧最优个体的时间为 t_{13} ,保留最优个体位置的时间为 t_{14} ,对最优个体的每一维边界处理的时间为 t_{15} ,则变异阶段的时间复杂度为:

$$T_4 = O(1 \times d \times (t_{12} + t_{13} + t_{15}) + t_{14} + 1 \times f(d)) = O(d + f(d))$$

综上所述,CHMPFA 的总时间复杂度为 $T = T_1 + T_{\max}(T_2 + T_3 + T_4) = O(d + f(d))$ 。因此,CHMPFA 的时间复杂度与标准 PFA 一致,执行效率并未下降。

3 实验仿真与结果分析

3.1 实验设计

为测试 CHMPFA 的性能,将 CHMPFA 与 PSO^[9], DE^[10], WOA^[11], BSA^[12], SCA^[13], ChOA^[14], BOA^[15] 以及标准 PFA^[1] 在 10 个基准测试函数(函数信息如表 1 所示)上进行 30 次独立实验,其中 $f_1 \sim f_5$ 为单峰函数(UN),有唯一的全局最优解,目的是检验算法的寻优能力和收敛速度; $f_6 \sim f_{10}$ 为多峰函数(MN),有较多的局部最优解,目的是验证算法跳出局部最优和避免过早收敛的能力。

为了实验的公正性,所有算法均在 Intel(R) Core (TM) i5—11260H CPU@ 2.60 GHz, Windows 10 系统,16 GB 内存,64 位操作系统的计算机上进行寻优测试,通过 MATLAB R2021a 软件进行仿真实验。统一设置实验种群规模 N 为 30,实验最大迭代次数 T_{\max} 为 1 000,每个算法在各基准测试函数上进行 30 次独立实验,将实验得到的平均值(Mean)与标准差(S.D)数据进行统计,实验参数如表 2 所示。

续表 3

Fun	Item	CHMPFA	PFA	PSO	WOA	DE	SCA	BOA	BSA	ChOA
f_5	Mean	3.42E-04	7.11E-02	3.83E+01	1.89E-03	2.19E+02	5.54E+01	9.65E-04	1.43E+03	3.61E-03
	SD	3.58E-04	8.11E-02	6.19E+01	2.46E-03	1.05E+02	4.60E+01	4.05E-04	2.28E+02	3.11E-03
	p -value	NaN(=)	3.34E-11(+)	3.02E-11(+)	1.86E-06(+)	3.02E-11(+)	3.02E-11(+)	4.44E-07(+)	3.02E-11(+)	1.21E-10(+)
f_6	Mean	-4.18E+04	-3.23E+04	-1.65E+04	-3.78E+04	-1.97E+04	-7.19E+03	-7.09E+03	-2.32E+02	-1.82E+04
	SD	5.89E+01	1.19E+04	2.79E+03	5.02E+03	2.96E+03	5.17E+02	7.79E+02	1.92E+03	9.94E+01
	p -value	NaN(=)	-1.14E-01(-)	3.02E-11(+)	5.53E-08(+)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	2.95E-11(+)
f_7	Mean	0	1.54E+02	7.83E+02	0	7.85E+02	2.15E+02	9.51E-13	1.48E+03	2.65E+00
	SD	0	6.22E+01	7.98E+01	0	1.65E+02	1.01E+02	2.46E-12	8.55E+01	4.50E+00
	p -value	NaN(=)	1.21E-12(+)	1.21E-12(+)	NaN(=)	1.21E-12(+)	1.21E-12(+)	3.13E-04(+)	1.21E-12(+)	1.21E-12(+)
f_8	Mean	8.88E-16	3.89E+00	4.41E+00	4.68E-15	2.00E+01	1.79E+01	3.38E-07	2.08E+01	2.00E+01
	SD	0	1.44E+00	1.03E+00	2.46E-15	5.45E-03	5.14E+00	5.99E-08	1.12E-01	3.31E-03
	p -value	NaN(=)	1.21E-12(+)	1.21E-12(+)	1.16E-09(+)	1.20E-12(+)	1.21E-12(+)	1.21E-12(+)	1.21E-12(+)	1.19E-12(+)
f_9	Mean	0	3.65E+00	1.23E+01	0	4.25E+02	4.62E+01	9.01E-10	2.93E+03	7.85E-03
	SD	0	1.53E+00	6.29E+00	0	4.59E+02	3.64E+01	5.38E-10	2.38E+02	2.05E-02
	p -value	NaN(=)	1.21E-12(+)	1.21E-12(+)	NaN(=)	1.21E-12(+)	1.21E-12(+)	1.21E-12(+)	1.21E-12(+)	1.21E-12(+)
f_{10}	Mean	1.25E+00	2.82E+01	2.03E+02	1.47E+00	1.54E+09	2.88E+08	9.98E+00	6.92E+09	9.35E+00
	SD	1.04E+00	2.39E+01	6.11E+02	4.92E-01	1.78E+09	2.09E+08	-5.63E-03	1.07E+09	1.74E-01
	p -value	NaN(=)	8.89E-10(+)	3.02E-11(+)	-7.73E-02(-)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)	3.02E-11(+)
MAE		9.96E+00	5.55E+03	1.22E+04	9.58E+04	1.54E+08	2.88E+07	2.29E+49	3.18E+53	4.08E+03
Rank		1	3	4	5	7	6	8	9	2

实验二:为了进一步评估 CHMPFA 的表现,通过 Wilcoxon 秩和检验^[16]来验证与其它算法之间是否存在显著的差异。该文选择 0.05 作为显著性水平进行 Wilcoxon 秩和检验,如果 CHMPFA 优于对比算法,则 p -value<0.05;如果 CHMPFA 与对比算法相当,则在数据中记录为 NaN;如果 CHMPFA 劣于对比算法,则 p -value \geq 0.05,并且将这些数据以黑体表示。该文使用“+/-/-”符号来表示 CHMPFA“优于/相当于/劣于”对比算法。

根据表 3 的实验结果可知,CHMPFA 与标准 PFA 在 9 个基准测试函数上得到的 p 值远小于显著性水平 0.05,表明 CHMPFA 在寻优性能上优于标准 PFA,这是因为 CHMPFA 集成了混沌反向学习初始化种群策略、赋予衰减性探路者更新方式和混合变异策略,Cat 混沌映射结合反向学习可以在初始阶段就引导种群朝着全局优势方向进行搜索,从而提高算法的全局优化性能;衰减因子帮助算法逐渐收敛到最优解附近,提高算法的稳定性和可靠性;对最优个体进行变异提高了算法跳出局部最优的概率,便于找到全局最优解。除探路者算法本身之外,CHMPFA 与除 WOA 外的 6 种算法在 10 个基准测试函数上得到的 p 值同样远小于显著性水平 0.05,说明 CHMPFA 较以上 6 种算法也有着更加优秀的寻优性能;而对于 WOA,即使存在着与 CHMPFA 性能相当或较劣的现象,但也是少数情况,

从整体算法性能看来,CHMPFA 仍占据显著优势。

实验三:基于 10 个基准测试函数的平均绝对误差 (Mean Absolute Error, MAE),将所有算法的性能定量分析后进行排序,是评估算法有效性与可行性的可靠方法^[17]。计算得到的 MAE 越小表示算法的平均结果与理论最优结果的绝对误差和越小,算法性能越优秀,其计算公式为:

$$MAE = \frac{\sum_{i=1}^{L_f} |\text{avg}_B B_i - \partial_i|}{L_f} \quad (19)$$

其中, $\text{avg}_B B_i$ 表示算法得到的全局最优解的平均值, ∂_i 表示选取基准测试函数的理论最优值, L_f 表示选取基准测试函数的个数。实验结果显示,CHMPFA 的 MAE 的排名第一,进一步验证了改进算法的鲁棒性。

3.3 收敛曲线分析

所有算法对函数求解寻优时的收敛曲线如图 2 所示(选取部分函数 f_2, f_5, f_6, f_8 和 f_9),为了方便观察曲线的收敛情况,对纵坐标取 10 为底的对数。

由图 2 可知,CHMPFA 在迭代前期收敛曲线下降较快,这是由于 PFA 初始阶段的种群由混沌反向学习策略生成,相较于随机生成种群,Cat 映射生成的种群遍布整个搜索空间,再通过反向学习策略,增加种群多样性的同时减少了个体重叠现象,更有利于探路者和跟随者的位置更新。

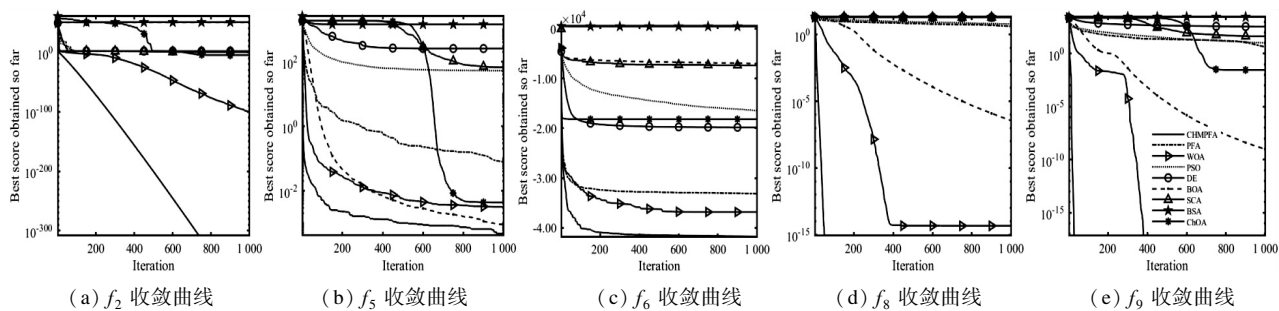


图 2 函数收敛曲线

在图 2 (a) 和图 2 (b) 的单峰函数图像中, CHMPFA 在 f_2 函数上收敛速度较快, 其余 8 种算法在寻优性能上较弱于 CHMPFA; 对于函数 f_5 , 虽然在迭代结束时 9 种算法均未能找到理论最优值, 但是 CHMPFA 的寻优精度相对较高, 收敛曲线有持续下降的趋势, 这是由于衰减因子为探路者在迭代后期缩减探索范围, 使探路者带领跟随者在局部区域能够更加精密的探索, 不断地向最优解区域靠近, 最终使得 CHMPF 在寻优精度上优于其它算法。

在图 2 (c) ~ (e) 的多峰函数图像中, CHMPFA 在 f_9 函数上 50 次迭代内就找到了理论最优值 0, 而除 WOA 外其余 7 种算法均出现了不同程度的停滞现象, 这是由于在 PFA 中加入了变异和贪婪选择策略, 以一定的概率选择高斯变异或者柯西变异的最优个体增强了跳出局部最优的能力, 该策略在 CHMPFA 寻优过程

中发挥着重要的作用; 在函数 f_6 和 f_8 上, 虽然所有算法均没有跳出局部区域, 但结合表 3 可知, CHMPFA 的迭代曲线收敛速最快, 在寻优精度上远超其它所对比的算法。

综上所述, 无论在单峰函数还是多峰函数, 在每个函数上 CHMPFA 的寻优精度和收敛速度都十分出色, 验证了综合三个改进策略 CHMPFA 的优化效果。

3.4 CEC2017 基准测试函数求解实验

为了进一步评估 CHMPFA 的鲁棒性, 在 CEC2017 基准函数中选取部分多峰函数、混合函数 (Hybrid Function, HF) 与复合函数 (Composition Function, CF) 进行求解寻优, 选取的函数如表 4 所示。设置初始种群规模为 30, 最大迭代次数为 1 000, 维度为 30, 各算法运行 30 次的平均值 (Mean) 与标准差 (S. D) 记录在表 5 中。

表 4 CEC2017 基准测试函数

函数	搜索范围	维度	f_{\min}	特征	函数	搜索范围	维度	f_{\min}	特征
CEC05	$[-100, 100]$	30	500	MN	CEC16	$[-100, 100]$	30	1 600	HF
CEC07	$[-100, 100]$	30	700	MN	CEC17	$[-100, 100]$	30	1 700	HF
CEC08	$[-100, 100]$	30	800	MN	CEC20	$[-100, 100]$	30	2 000	CF
CEC09	$[-100, 100]$	30	900	MN	CEC21	$[-100, 100]$	30	2 100	CF
CEC13	$[-100, 100]$	30	1 300	HF	CEC24	$[-100, 100]$	30	2 400	CF
CEC15	$[-100, 100]$	30	1 500	HF	CEC29	$[-100, 100]$	30	2 900	CF

表 5 CEC2017 基准测试函数寻优结果对比

Fun	Item	CHMPFA	PFA	PSO	WOA	DE	SCA	BSA	BOA	ChOA
CEC05	Mean	6.20E+02	6.68E+02	7.64E+02	8.33E+02	6.80E+02	8.18E+02	1.53E+03	9.09E+02	8.29E+02
	S. D	2.58E+01	3.52E+01	3.50E+01	5.54E+01	4.30E+01	2.53E+01	2.09E+02	2.43E+01	3.42E+01
CEC07	Mean	8.90E+02	9.38E+02	1.10E+03	1.29E+03	1.25E+03	1.20E+03	4.98E+03	1.40E+03	1.25E+03
	S. D	3.36E+01	6.22E+01	4.56E+01	8.65E+01	1.83E+02	5.00E+01	8.63E+02	3.95E+01	5.42E+01
CEC08	Mean	9.11E+02	9.19E+02	1.06E+03	1.07E+03	9.76E+02	1.20E+03	1.71E+03	1.13E+03	1.08E+03
	S. D	2.44E+01	4.77E+01	3.73E+01	5.65E+01	5.97E+01	5.10E+01	1.43E+02	1.84E+01	1.84E+01
CEC09	Mean	2.82E+03	3.58E+03	6.25E+03	1.09E+04	8.38E+03	7.68E+03	3.92E+04	1.04E+04	8.48E+03
	S. D	1.10E+03	1.60E+03	2.32E+03	3.81E+03	2.98E+03	1.80E+03	1.17E+04	1.14E+03	1.52E+03
CEC13	Mean	1.23E+05	1.24E+05	4.03E+08	1.90E+06	3.26E+08	9.21E+08	1.03E+11	8.75E+09	4.56E+09
	S. D	6.22E+04	9.13E+04	1.12E+09	1.76E+06	1.40E+09	3.62E+08	4.06E+10	4.19E+09	3.98E+09

续表 5

Fun	Item	CHMPFA	PFA	PSO	WOA	DE	SCA	BSA	BOA	ChOA
CEC15	Mean	3.83E+04	5.00E+04	5.56E+05	1.53E+06	7.09E+07	4.54E+07	4.04E+10	3.94E+08	1.37E+07
	S. D	2.82E+04	4.19E+04	6.11E+05	1.95E+06	3.33E+08	2.95E+07	1.62E+10	2.68E+08	1.20E+07
CEC16	Mean	2.82E+03	3.03E+03	3.72E+03	4.05E+03	3.50E+03	3.96E+03	7.31E+04	7.72E+03	4.22E+03
	S. D	2.84E+02	3.67E+02	4.44E+02	4.41E+02	6.01E+02	3.18E+02	5.69E+04	1.76E+03	4.38E+02
CEC17	Mean	2.18E+03	2.41E+03	2.52E+03	2.75E+03	2.72E+03	2.66E+03	1.76E+07	7.14E+03	2.92E+03
	S. D	1.78E+02	2.35E+02	2.82E+02	2.34E+02	2.81E+02	1.98E+02	3.02E+07	4.75E+03	4.01E+02
CEC20	Mean	2.50E+03	2.76E+03	2.82E+03	2.89E+03	3.07E+03	2.90E+03	4.58E+03	2.89E+03	3.06E+03
	S. D	1.52E+02	2.18E+02	1.93E+02	2.47E+02	2.38E+02	2.42E+02	3.53E+02	1.11E+02	1.71E+02
CEC21	Mean	2.41E+03	2.45E+03	2.57E+03	2.63E+03	2.48E+03	2.62E+03	3.73E+03	2.66E+03	2.60E+03
	S. D	2.92E+01	3.81E+01	4.19E+01	5.88E+01	5.18E+01	6.08E+01	8.55E+02	1.33E+02	3.24E+01
CEC24	Mean	2.93E+03	3.01E+03	3.65E+03	3.24E+03	3.02E+03	3.23E+03	6.65E+03	4.01E+03	3.30E+03
	S. D	3.08E+01	5.94E+01	2.32E+02	1.00E+02	6.98E+01	4.62E+01	1.30E+03	4.05E+02	6.02E+01
CEC29	Mean	4.27E+03	4.41E+03	4.93E+03	5.35E+03	4.55E+03	5.10E+03	2.81E+08	9.03E+03	4.79E+03
	S. D	1.95E+02	3.12E+02	1.03E+03	4.29E+02	3.38E+02	2.29E+02	9.14E+08	3.03E+03	2.78E+02

根据表 5 可知,CHMPFA 在 12 个 CEC2017 基准测试函数上寻优所得的平均值和标准差均比其余算法的优,这是由于 CHMPFA 引入了混沌反向学习策略,赋予种群更深层次的多样性,使初始种群不再随机地遍布于搜索空间;并将衰减因子引入探路者公式中,帮助探路者更快地遍历解空间,通过逐渐减小搜索步长从而加深对局部区域的开采,提高了找到全局最优解的概率;由于 CEC2017 函数局部极值众多,所以混合变异策略在 CHMPFA 寻优过程中发挥着重要的作用,使算法能够跳出局部极值向函数理论最优值靠近。

4 压力容器设计问题

压力容器设计问题的目标是使各项费用总和最低。这一模型有 4 个决策变量:筒体厚度 T_s ($0 \leq x_1 \leq 99$)、封头厚度 T_h ($0 \leq x_2 \leq 99$)、筒体半径 R ($0 \leq x_3 \leq$

200)和圆柱形截面长度 L_s ($10 \leq x_4 \leq 200$),以及 4 个约束条件,压力容器设计问题数学模型如下:

$$\begin{aligned}
 \min f(X) &= 0.622 \, 4x_1x_2x_3 + 1.778 \, 1x_2x_3^2 + \\
 &\quad 3.166 \, 1x_1^2x_4 + 19.84x_1^2x_3 \\
 \text{s. t. } g_1(X) &= -x_1 + 0.019 \, 3x_2 \leq 0 \\
 g_2(X) &= -x_2 + 0.009 \, 54x_3 \leq 0 \\
 g_3(X) &= -\pi x_3^2x_4 - \frac{4\pi x_3^3}{3} + 1 \, 296 \, 000 \leq 0 \\
 g_4(X) &= x_4 - 240 \leq 0
 \end{aligned} \quad (20)$$

表 6 记录了 9 种算法优化压力容器设计的结果,CHMPFA 获得的最优解为 $[T_s, T_h, R, L_s] = [0.778 \, 2, 0.384 \, 65, 40.319 \, 6, 199.999 \, 7]$,最优值 $f(X) = 5 \, 885.334 \, 5$,CHMPFA 比其它 8 种元启发式算法的优化效果出色,有较低的设计成本,在压力容器设计问题中起到了良好的优化效果。

表 6 压力容器设计问题最优方案

算法	x_1	x_2	x_3	x_4	$f(X)$
CHMPFA	0.778 2	0.384 7	40.319 6	199.999 7	5 885.334 5
PFA	0.778 2	0.384 7	40.322 4	199.961 1	5 885.425 0
PSO	0.000 0	0.000 0	40.320 9	200.000 0	7 590.545 7
WOA	0.801 7	0.404 8	41.859 2	179.624 0	5 986.777 6
DE	0.790 2	0.390 6	40.941 6	191.595 8	5 907.852 5
SCA	1.269 9	0.000 0	65.356 5	10.839 7	7 694.517 5
BSA	13.685 8	12.587 2	61.485 4	19.416 5	426 520.269 2
BOA	1.091 0	0	57.700 1	48.562 5	8 123.176 2
ChOA	1.253 5	0	64.634 9	12.625 7	7 647.346 6

5 结束语

为提升标准 PFA 的收敛速度和寻优精度,提出了一种嵌入 Cat 映射的混合变异探路者算法(CHMPFA)。通过仿真实验与其它8种具有代表性的智能优化算法进行性能对比,验证 CHMPFA 多种策略的有效性,并将 CHMPFA 应用于压力容器问题中,其良好的优化效果和稳定性表明了 CHMPFA 的鲁棒性。未来考虑应用 CHMPFA 解决多目标优化问题或更为复杂的实际问题。

参考文献:

- [1] YAPICI H, CETINKAYA N. A new meta-heuristic optimizer: pathfinder algorithm[J]. Applied Soft Computing, 2019, 78: 545–568.
- [2] 逮 苗. 灰狼优化算法的改进及应用[D]. 南宁: 广西民族大学, 2021.
- [3] TANG C, ZHOU Y, LUO Q, et al. An enhanced pathfinder algorithm for engineering optimization problems[J]. Engineering with Computers, 2022, 38(2): 1481–1503.
- [4] TANG C, ZHOU Y, TANG Z, et al. Teaching-learning-based pathfinder algorithm for function and engineering optimization problems[J]. Applied Intelligence, 2021, 51(7): 5040–5066.
- [5] 孙哲中. 改进的探路者算法及其应用[D]. 鞍山: 辽宁科技大学, 2022.
- [6] 徐辰华, 李成县, 喻 昕, 等. 基于 Cat 混沌与高斯变异的改进灰狼优化算法[J]. 计算机工程与应用, 2017, 53(4): 1–9.
- [7] LI W, WANG G G, GANDOMI A H. A survey of learning-based intelligent optimization algorithms[J]. Archives of Computational Methods in Engineering, 2021, 28: 3781–3799.
- [8] 李爱莲, 全凌翔, 崔桂梅, 等. 融合正余弦和柯西变异的麻雀搜索算法[J]. 计算机工程与应用, 2022, 58(3): 91–99.
- [9] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95–international conference on neural networks. Perth: IEEE, 1995: 1942–1948.
- [10] DRAA A, CHETTAH K, TALBI H. A compound sinusoidal differential evolution algorithm for continuous optimization[J]. Swarm and Evolutionary Computation, 2019, 50(C): 100450.
- [11] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51–67.
- [12] 邵良杉, 韩瑞达. 基于天牛须搜索的花朵授粉算法[J]. 计算机工程与应用, 2018, 54(18): 188–194.
- [13] MIRJALILI S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge – Based Systems, 2016, 96: 120–133.
- [14] KHISHE M, MOSAVI M R. Chimp optimization algorithm[J]. Expert Systems with Applications, 2020, 149: 113338.
- [15] 王依柔, 张达敏, 徐 航, 等. 基于自适应扰动的疯狂蝴蝶算法[J]. 计算机应用研究, 2020, 37(11): 3276–3280.
- [16] WILCOXON F. Individual comparisons by ranking methods[J]. Biometrics Bulletin, 1945, 1(6): 80–83.
- [17] NABIL E. A modified flower pollination algorithm for global optimization[J]. Expert Systems with Applications, 2016, 57: 192–203.