

## 求解高维复杂函数的混合蛙跳-灰狼优化算法

黄晨晨, 魏霞<sup>†</sup>, 黄德启, 叶家豪

(新疆大学 电气工程学院, 新疆 乌鲁木齐 830047)

**摘要:** 针对高维复杂函数问题, 提出一种混合蛙跳-灰狼优化算法(SFL-GWO). 该算法通过改进的Logistic映射初始化GWO算法种群提高算法的多样性; 其次, 提出一种新的距离控制参数的非线性调整策略来增强种群的探索与开发的能力; 最后通过引入改进的随机蛙跳算法中改变最差位置的方式使SFL-GWO算法跳出局部最优的局限. 通过选取的10个高维复杂函数的寻优结果验证了算法的性能, 并与粒子群优化算法(PSO)、灰狼优化算法(GWO)和鲸鱼优化算法(WOA)3种基本算法以及与8种改进算法的寻优的结果进行了比较. 仿真结果证明: SFL-GWO算法在不仅可以提高收敛精度也可以提高算法的搜索速度, 证明了SFL-GWO算法在求解高维复杂函数的高效性.

**关键词:** 高维复杂函数; 改进的Logistic映射; 距离控制参数; 灰狼优化算法; 混合蛙跳算法

**引用格式:** 黄晨晨, 魏霞, 黄德启, 等. 求解高维复杂函数的混合蛙跳-灰狼优化算法. 控制理论与应用, 2020, 37(7): 1655 – 1666

DOI: 10.7641/CTA.2020.90461

## Shuffled frog leaping grey wolf algorithm for solving high dimensional complex functions

HUANG Chen-chen, WEI Xia<sup>†</sup>, HUANG De-qi, YE Jia-hao

(College of Electrical Engineering, Xinjiang University, Urumqi Xinjiang 830047, China)

**Abstract:** Aiming at the problem of high-dimensional complex function, a shuffled frog leaping-grey wolf optimization algorithm (SFL-GWO) is proposed. Firstly, in order to improve species diversity of the algorithm, the improved Logistic map is used to initialize the GWO algorithm. Secondly, a new nonlinear adjustment strategy of distance control parameters is proposed to enhance the population exploration and development ability. Thirdly, an improved random strategy of Frog Leaping is introduced, which can change the worst position to let SFL-GWO algorithm jumping out the local optimum. Finally, the performance of the algorithm is verified by the optimization results of 10 high-dimensional complex functions. The performance results demonstrated that the SFL-GWO algorithm can not only improve the convergence accuracy but also improve the search speed of the algorithm in solving high-dimensional complex functions compared with particle swarm optimization (PSO), grey wolf optimization (GWO), wolf optimization algorithm (WOA) and several improved algorithms.

**Key words:** high-dimensional complex function; improved Logistic mapping; distance control parameter; grey wolf optimization algorithm; shuffled frog leaping algorithm

**Citation:** HUANG Chenchen, WEI Xia, HUANG Deqi, et al. Shuffled frog leaping grey wolf algorithm for solving high dimensional complex functions. *Control Theory & Applications*, 2020, 37(7): 1655 – 1666

### 1 引言

在实际的工程问题中高维复杂函数问题的优化一直以来都是十分重要的问题, 诸如大型电子系统设计、大量资源的调度、大规模路由器网络、化学动力学、大规模车间调度问题<sup>[1]</sup>、热连轧<sup>[2]</sup>和冷连轧厚度负荷分配等科学和工程问题都可以用高维复杂函数表示, 有效地求解高维复杂函数具有很重要的研究价

值. 高维复杂函数多数情况下指的是函数维度超过100维, 具有非线性、高复杂度的特点, 随着维数的增加, 函数的复杂度会呈指数级的速度增长, 因此求解具有很大的困难性. 目前针对于求解高维复杂函数的优化问题, 徐东方等人<sup>[3]</sup>提出一种关于多种群子空间学习粒子群算法 (particle swarm optimization, PSO), 通过多种群的进化来提高PSO算法收敛的速度. Omi-

收稿日期: 2019-06-18; 录用日期: 2020-02-16.

<sup>†</sup>通信作者. E-mail: 30462111@qq.com; Tel.: +86 13319811560.

本文责任编辑: 王凌.

国家自然科学基金项目(51468062)资助.

Supported by the National Natural Science Foundation of China (51468062).

dvar等人<sup>[4]</sup>提出了一种自动分解的策略对大规模优化问题进行求解, 有效地对分解了问题降低求解的复杂性. 胡成玉等人<sup>[5]</sup>提出了一种基于非静态的维度交叉改进PSO算法的策略解决在求解高维复杂函数时易陷入早熟的问题. 田瑾<sup>[6]</sup>通过对量子粒子群优化(quantum particle swarm optimization, QPSO)算法的进一步改进, 大大提高了PSO收敛精度以及收敛的速度. 于万霞等人<sup>[7]</sup>利用遗传算法(genetic algorithm, GA)和PSO算法相结合, 提出了关于主从结构的算法, 很好地提高了在求解过程中的全局探索能力以及局部优化的能力. 以上改进算法在全局收敛性和求解精度方面有了进一步地提高但是由于算法本身搜索机制的局限, 对于一些高维复杂函数的求解精度还是有一定的局限性.

灰狼优化算法(grey wolf optimizer, GWO)是一种在2014年被Mirjalili等人<sup>[8]</sup>提出的模拟灰狼群体中的捕食行为、社会等级机制的元启发式群体智能优化算法, GWO算法有简单地模型结构、控制参数少, 算法收敛速度快、寻优性能良好的特点, 但是也有着易陷入收敛早熟、迭代后期收敛慢和求解精度低的缺点. GWO算法被提出以来有很多的研究人员对其进行改进, 利用佳点集策略<sup>[9]</sup>、混沌序列策略<sup>[10]</sup>生成初始种群, 控制参数的非线性调整<sup>[11]</sup>等. 顾清华等人<sup>[12]</sup>提出了一种混合了遗传算法和灰狼优化算法的策略, 提高了算法的收敛精度. 龙文等人<sup>[13]</sup>提出一种利用光学透镜成像原理的反向学习策略来解决GWO算法在迭代是易陷入局部最优的问题. 滕志军等人<sup>[14]</sup>提出一种基于Tent映射的混合灰狼优化算法, 引入PSO算法中将个体自身经历过最优值与种群最优值相结合的思想来更新灰狼个体的位置信息, 解决算法在求解过程中鲁棒性差的问题, 提高了算法的收敛速度. 姜天华<sup>[15]</sup>通过引入改进收敛因子非线性调整策略以及带权重系数的个体位置更新方法, 加入局部搜索算法, 用于加强局部搜索能力. 龙文、蔡绍洪等人<sup>[16]</sup>提出一种基于混沌和精英反向学习的混合灰狼优化算法以解决高维优化问题提高算法在求解过程中收敛速度慢和收敛精度低的问题. MING Z F等人<sup>[17]</sup>通过引入变异算子和淘汰重组机制来改进GWO算法提高了算法的搜索速度以及收敛精度. 虽然GWO算法虽然有了许多的改进, 但是在求解高维复杂函数的问题上, 灰狼算法的应用还是不多. 随着函数的维度增加, 高维问题的复杂性和求解的困难也会增加很大, 高维函数的求解精度也会有所下降. 为了解决在求解高维复杂函数中遇到的问题, 本文采用Logistic映射<sup>[18]</sup>产生初始化灰狼种群保持种群的多样性; 通过调整距离控制参数 $a$ 为非线性递减来改变GWO算法的探索 and 开发能力; 引入随机蛙跳算法<sup>[19]</sup>的最差位置改变式控制灰狼算法跳出局部最优.

## 2 基本灰狼算法

在灰狼种群中,  $\alpha$ 狼的位置代表了最优位置对应着求解问题的最优解,  $\beta$ 狼的位置代表了第2优位置对应着求解问题的次优解,  $\delta$ 狼的位置表示第3优位置对应着第3优解,  $\omega$ 狼的位置则对应着解空间中的剩余候选解. 在围捕猎物的过程中,  $\omega$ 狼通过跟随 $\alpha$ ,  $\beta$ ,  $\delta$ 狼的位置逐步更新自己的位置, 这也是求解最优解的过程. 假设当灰狼种群规模为 $N$ , 解集空间为 $D$ 维, 那么有 $X_i^d(t+1)$ 的更新式如下:

$$B = |C \cdot X_p^d(t) - X_i^d(t)|, \quad (1)$$

$$X_i^d(t+1) = X_i^d(t) - A \cdot B, \quad (2)$$

其中: 猎物的位置表示为 $X_p$ ,  $X_i^d(t)$ 为第 $i$ 只灰狼当前的位置,  $t$ 为到目前为止的迭代次数,  $A$ 表示收敛因子控制灰狼个体趋向或者远离当前猎物,  $C$ 表示模拟灰狼在围捕猎物的过程中所受到的自然界的阻碍作用.

同时 $A$ ,  $C$ 的计算式如下:

$$A = 2a \cdot r_1 - a, \quad (3)$$

$$C = 2r_2, \quad (4)$$

$$a = 2 - \frac{2t}{T_{\text{iteration}}}, \quad (5)$$

其中:  $A$ 的值是区间 $[-2a, 2a]$ 之间的值,  $A$ 随着 $a$ 的变化而变化, 而 $C$ 的值为区间 $[0, 2]$ 间的随机值,  $r_1$ 与 $r_2$ 是 $[0, 1]$ 间的随机数,  $a$ 从2开始线性递减到0. 值得注意的, 当 $|A| > 1$ 表示灰狼离开当前猎物去寻找更有希望的猎物, 即更有希望的最优解; 当 $|A| < 1$ 表示灰狼个体开始攻击当前的猎物, 即开始移动到当前解.  $C > 1$ 或 $C < 1$ 表示随机加强或减弱猎物距离的作用. 其中, 灰狼种群中的其他灰狼个体 $X_i$ 通过 $\alpha$ ,  $\beta$ ,  $\delta$ 狼的当前位置 $X_\alpha$ ,  $X_\beta$ ,  $X_\delta$ 来改变自身的位置, 改变式(6)和式(7):

$$\begin{cases} X_{i,\alpha}^d(t+1) = X_\alpha^d(t) - A|C \cdot X_\alpha^d(t) - X_i^d(t)|, \\ X_{i,\beta}^d(t+1) = X_\beta^d(t) - A|C \cdot X_\beta^d(t) - X_i^d(t)|, \\ X_{i,\delta}^d(t+1) = X_\delta^d(t) - A|C \cdot X_\delta^d(t) - X_i^d(t)|, \end{cases} \quad (6)$$

$$X_i^d(t+1) = \frac{X_{i,\alpha}^d(t) + X_{i,\beta}^d(t) + X_{i,\delta}^d(t)}{3}. \quad (7)$$

## 3 混合蛙跳-灰狼优化算法

### 3.1 混沌系统初始化种群方法

为了使初始种群中的个体更好地遍及整个的解空间以及提高混合蛙跳-灰狼优化算法(shuffled frog leaping-grey wolf optimizer, SFL-GWO)的求解高维复杂函数问题时的搜索效率. 本文采用改进的Logistic混沌映射来进行种群的初始化, 改进的Logistic混沌映射可以扩大控制参数选择区域; 使映射区域近似于满映射状态; 产生的混沌系统有良好的均匀分布的特点, 克服了Logistic映射<sup>[20]</sup>的存在无穷不动点、稳定

窗与空白区问题、不易实现满映射的状态的局限性,改进后的Logistic混沌映射保留了标准Logistic映射的复杂的动力学特性,改进后的Logistic混沌映射具有较好的随机性、规律性和遍历性.其数学模型为

$$x_{n+1} = \begin{cases} 4x_n(1-x_n), & G, \\ 1-\mu x_n(1-x_n), & \text{其他}, \end{cases} \quad (8)$$

其中:

$$G = \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{|\mu|}{4}} \leq x_n \leq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{|\mu|}{4}},$$

$\mu$ 为系统的参数,  $0 < \mu < 1$ . 通过验证本文设置初始值  $x_0 = 0.001, \mu \in [0, 4]$ .

### 3.2 距离控制参数的改进

由式(3)可以看出,  $|A| > 1, |A| < 1$  直接影响了算法的探索与开发能力,而系数 $|A|$ 的变化受距离控制参数 $a$ 的影响. 但是由式(5)可知 $a$ 是从2线性递减到0的. 在迭代前期,较大的距离控制参数 $a$ 值可以使得算法具有较大搜索步长,提高探索能力较强,避免算法出现早熟收敛的情况;在迭代后期,较小的距离控制参数 $a$ 值使得群体集中在某个区域内展开搜索,提高算法的开发能力,加快算法的收敛速度. 然而,在高维复杂函数优化问题中,由于算法搜索过程极为复杂,控制参数 $a$ 线性递减策略很难满足搜索的实际情况. 因此,本文提出的SFL-GWO算法改进了原来的线性递减的距离控制参数为非线性递减,改进的距离控制参数式(9):

$$a = (a_{\text{initial}} - a_{\text{final}}) \times \left(1 - \sin\left(\left(\frac{t}{T_{\text{iteration}}}\right)^2 \cdot \frac{\pi}{2}\right)\right), \quad (9)$$

其中: 距离控制参数的初始值  $a_{\text{initial}} = 2$ , 最终的距离控制参数为  $a_{\text{final}} = 0$ ,  $a$ 是从2到0非线性递减的.

为了验证本文提出的距离控制参数的有效性,将提出的距离控制参数以及文献[21, 22]中的改进策略进行非线性的对比. 其中 $a_1, a_2, a_3, a_4, a_5$ 分别表示有

$$a_1 = (a_{\text{initial}} - a_{\text{final}}) \times \left(1 - \sin\left(\left(\frac{t}{T_{\text{iteration}}}\right)^2 \cdot \frac{\pi}{2}\right)\right), \quad (10)$$

$$a_2 = a_{\text{initial}} - (a_{\text{initial}} - a_{\text{final}}) \times \frac{t}{T_{\text{iteration}}}, \quad (11)$$

$$a_3 = a_{\text{initial}} - (a_{\text{initial}} - a_{\text{final}}) \times \frac{1}{e-1} \times \left(e^{\frac{t}{T_{\text{iteration}}}} - 1\right), \quad (12)$$

$$a_4 = (a_{\text{initial}} - a_{\text{final}}) \times \cos\left(\frac{t}{T_{\text{iteration}}}\right), \quad (13)$$

$$a_5 = a_{\text{initial}} - (a_{\text{initial}} - a_{\text{final}}) \times \left(\frac{t}{T_{\text{iteration}}}\right)^2. \quad (14)$$

为了更加直观的看出改进后的效果,本文对5种距离控制参数进行对比仿真实验如图1. 可以看出,本文

提出的新的非线性距离控制参数,在算法搜索前期具有较大的值来增强算法探索的能力,在算法搜索后期具有较小的距离控制参数来实现对某个区域的开发. 综上,新提出的非线性距离控制参数可以更加高效地来平衡局部搜索以及全局搜索的能力.

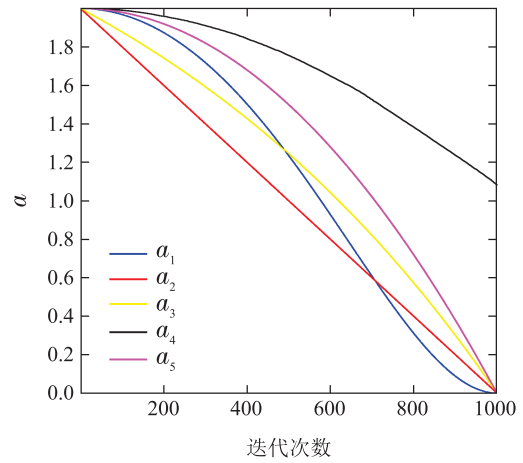


图 1 距离控制参数对比曲线

Fig. 1 Comparison curves of distance control parameter

### 3.3 改进SFLA最差位置改变策略

GWO算法在位置的迭代更新的过程中只考虑了每个灰狼个体位置信息和最优、次优以及第3优的位置信息,以及实现灰狼个体和种群间的信息交流,但是没有考虑如果陷入局部最优后如何跳出局部最优的策略,因此,引入混合蛙跳最差位置改变的策略来改进灰狼种群的位置更新信息.

随机蛙跳算法(shuffled frog leaping algorithm, SF-LA)在2003年被Eusuff等人<sup>[19]</sup>提出的,经过研究发现SFLA算法具有结构简单、设置参数少、鲁棒性强等优点从而被广泛的应用在水资源网络优化问题、成品油管网优化问题等工程领域.

引入一种改进的SFLA算法<sup>[23]</sup>中的最差蛙的位置改变策略可以有效地防止GWO算法陷入早熟收敛的问题. 首先,计算灰狼个体位置对应的适应度值,将最差适应度值所对应的灰狼的位置按照式(15)调整获得新的灰狼位置,表达式如下:

$$X_i^d(t+1) = X_i^d(t) + \text{rand}(\cdot) \cdot (X_\alpha^d(t) - X_\gamma^d(t)), \quad (15)$$

其中:  $\text{rand}(\cdot)$ 为[0, 1]之间的随机数,  $X_\alpha$ 为 $\alpha$ 的位置,  $X_\gamma$ 为最差适应度对应的 $\gamma$ 的位置.

通过加入改变概率  $P_m = 0.05$ , 在每次迭代的过程中对比迭代时设置的随机数  $r_3 = \text{rand}()$  和的值来判断是否需要根据改进随机蛙跳算法(improved shuffled frog leaping algorithm, ISFLA)算法的最差位置改变式(15)来让SFL-GWO算法中的最差灰狼个体改变自身的位置跳出当前的解去寻找更合  $P_m$  适的解. 当

$r_3 < P_m$ 时,让最差的 $\gamma$ 狼根据式(15)来改变自身的最差位置,去寻找更合适的解;否则, $\gamma$ 狼依旧保持着最差的位置.

### 4 算法流程

综上所述,本文提出的SFL-GWO算法的基本流程如下:

**步骤1** 设置算法参数:种群数量规模 $N = 50$ ,最大迭代次数 $T_{iteration} = 1000$ ,距离控制参数初值 $a_{initial} = 2$ 、终值 $a_{final} = 0$ ,最差位置改变概率 $P_m = 0.05$ .

**步骤2** 利用Logistic混沌映射初始化产生灰狼种群 $\{f(X_i), i = 1, 2, \dots, N\}$ ,并记录当前的最优个体 $\alpha$ 、第2优个体 $\beta$ 和第3优个体 $\delta$ ,并新定义一个灰狼个体记录当前迭代次数中的最差个体 $\gamma$ ,其对应位置 $X_\alpha, X_\beta, X_\delta$ 和 $X_\gamma$ .

**步骤3** 算法迭代,根据式(3)-(4)(9)计算距离控制参数 $a$ 以及参数 $A, C$ ;同时按照概率 $P_m$ 随机选择个体按照式(15)跳出局部最优位置;计算灰狼种群的适

应度值 $\{f(X_i), i = 1, 2, \dots, N\}$ ,更新最优个体 $\alpha$ 狼、第2优个体 $\beta$ 狼和第3优个体 $\delta$ 狼以及最差个体 $\gamma$ 狼,其对应位置 $X_\alpha, X_\beta, X_\delta$ 和 $X_\gamma$ .

**步骤4** 当灰狼种群找到最优值或迭代达到迭代最大次数后算法结束,并记录当前搜索到的最优值.

### 5 仿真实验及数据分析

#### 5.1 测试函数

为了验证SFL-GWO算法的寻优性能效果,选取了GWO算法<sup>[8]</sup>中提到的13个测试函数(30维)对比测试GWO、混合遗传灰狼算法(hybrid genetic grey wolf algorithm, HGGWO)以及SFL-GWO的寻优性能,以验证改进后的算法的有效性.此外,为了更好地验证改进后的算法对求解高维复杂函数的能力,选取10个高维测试函数对PSO、鲸鱼优化算法(whales optimization algorithm, WOA)、GWO和SFL-GWO进行寻优测试;同时为了更加全面的证明SFL-GWO算法的收敛精度高,将SFL-GWO算法与8种改进算法进行寻优性能的比较分析.其中,选择的标准测试函数见表1.

表 1 10个测试函数  
Table 1 10 test functions

函数名	函数表达式	范围	$f_{min}$	收敛精度
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	$[-100, 100]$	0	1E-8
Schwefel's2.22	$f_2(x) = \sum_{i=1}^d  x_i  + \prod_{i=1}^d  x_i $	$[-10, 10]$	0	1E-8
Schwefel's2.26	$f_3(x) = \sum_{i=1}^d -x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	-418.9829	1E-2
Zakharov	$f_4(x) = \sum_{i=1}^d x_i^2 + (\sum_{i=1}^d 0.5ix_i)^2 + (\sum_{i=1}^d 0.5ix_i)^4$	$[-5, 10]$	0	1
Powell	$f_5(x) = \sum_{i=1}^{\frac{d}{4}} ((x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + x_{4i})^2 + (x_{4i-2} + 2x_{4i-1})^4 + 10(x_{4i-3} + x_{4i})^4)$	$[-4, 5]$	0	1E-3
Quartic	$f_6(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0	1E-4
Rastrigin	$f_7(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0	1E-8
Ackley	$f_8(x) = -20 \exp(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}) - \exp(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)) + 20$	$[-32, 32]$	0	1E-5
Griewank	$f_9(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0	1E-5
Penalized 1	$f_{10}(x) = \frac{\pi}{d} \{10 \sin(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 (1 + \sin^2(\pi y_i + 1)) + (y_d - 1)^2\} + \sum_{i=1}^d u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) \begin{cases} k(x_i - a)^m, & x_i > a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$	0	1E-2

### 5.2 仿真结果及分析

为了保证寻优过程的客观性, 设置种群规模为  $N = 50$ , 迭代次数  $T_{iteration} = 1000$ . 此外, 在PSO中, 算法参数为  $c_1 = c_2 = 2$ ,  $w = 0.7294$ ; 在GWO中, 设置参数  $a$  的初始值为2, 参数  $a$  的终值为0; SFL-GWO算法中, 设置随机跳出的概率  $P_m = 0.05$ . 将所有算法都独立地测试30次, 记录算法测试的结果、平均值以及标准差. 所有的测试都在Intel Core i3-7100,

4 GB内存, 3.90 GHZ的计算机上运行, 编写的MATLAB代码程序均在MATLAB R2017b上实现.

#### 5.2.1 混沌系统初始化种群仿真分析

采用改进的Logistic混沌映射对GWO算法的种群进行初始化可以有效地提高GWO算法的种群多样性、搜索效率以及避免算法陷入局部最优. 如图2表示了测试函数为30维时, 分别对单峰函数  $f_4$ ,  $f_6$ , 多峰函数  $f_7$ ,  $f_{10}$  寻优的结果.

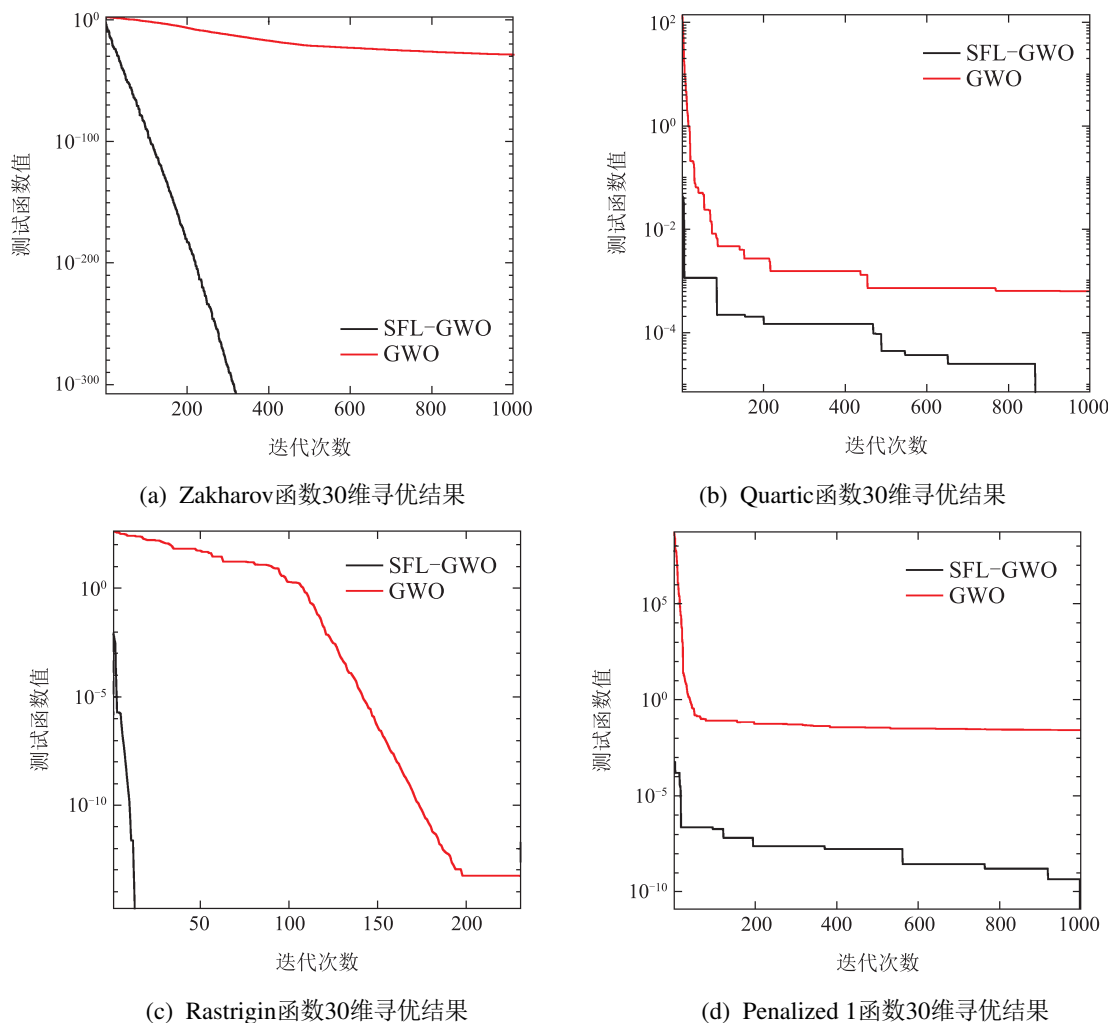


图 2 加入改进的Logistic混沌映射的寻优结果

Fig. 2 Optimization results of an improved Logistic chaotic map are added

通过图2可以看出, 在单独加入改进的Logistic混沌映射可以提高GWO算法的寻优效果, 通过实验验证单独加入改进的Logistic混沌映射可以找到Zakharov函数的最优值0, 对于Quartic函数所找到的最优值为  $7.31E-06$ , 在Rastrigin函数的寻优结果上也可以找到最优值0, 在对于函数Penalized 1的寻优结果也是可以达到函数的收敛精度. 再通过对其他的测试函数进行仿真实验验证, 可以发现对于部分测试函数的寻优结果并不是很理想. 最后可以得出

结论, 单独地加入改进的Logistic混沌映射可以找到部分测试函数的最优值, 但对于部分多峰函数不能够完全的使算法跳出局部最优.

#### 5.2.2 距离控制参数 $a$ 的仿真分析

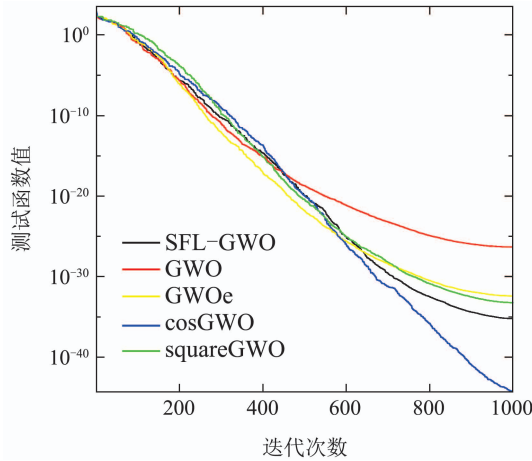
为了验证改进的距离控制参数  $a$  的有效性, 分别用  $a_1, a_2, a_3, a_4, a_5$  来代替原来的距离控制参数, 改变了距离参数后的算法分别为SFL-GWO, GWO, GWOe, cosGWO和squareGWO, 为了更加直观的验证改进的距离参数的有效性, 如图3给出了在测



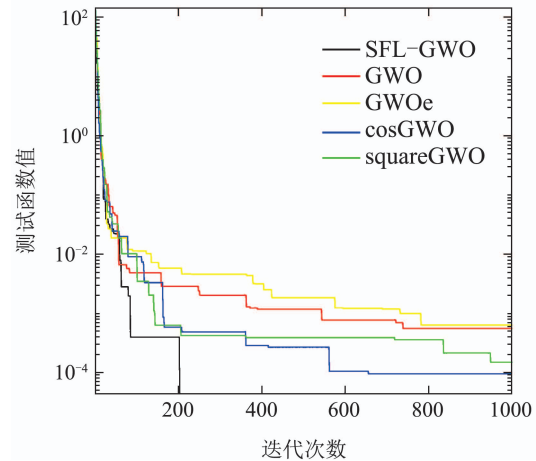
试函数为30维的情况下,分别对单峰函数 $f_4, f_6$ , 多峰函数 $f_7, f_{10}$ 寻优曲线图.

通过图3分析可知,改进的距离控制参数对于测试函数可以在一定程度上提高算法的探索与开发的能力,但是依旧存在着无法跳出局部最优的情况,

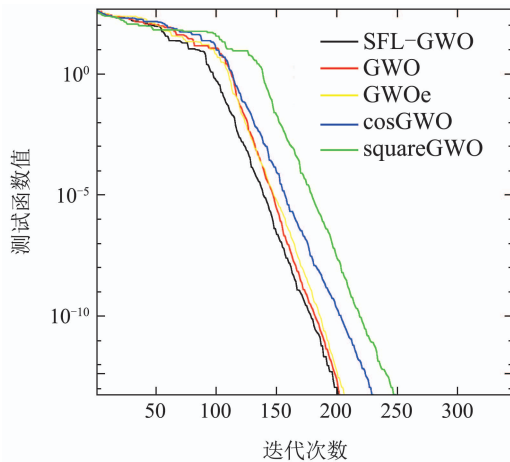
对于单峰函数Zakharov和Quartic来说,算法可以提高搜索速度以及搜索精度但是效果并不明显;对于多峰函数Rastrigin和Penalized 1来说,改进的距离控制参数可以适当提高搜索速度但是还是很容易陷入局部最优的问题.



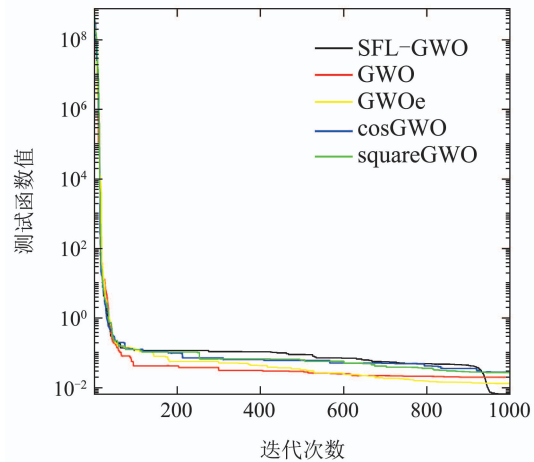
(a) Zakharov函数30维寻优结果



(b) Quartic函数30维寻优结果



(c) Rastrigin函数30维寻优结果



(d) Penalized 1函数30维寻优结果

图3 不同的距离参数的寻优结果

Fig. 3 Optimization results of different distance parameters

### 5.2.3 加入混合蛙跳算法的仿真分析

通过加入改进的Logistic混沌映射初始化算法种群、改变控制参数 $a$ 可以找到表1中的部分测试函数的最优值,对于剩余的测试函数还是无法找到最优值.通过分析可以知道,对于无法找到最优值的测试函数,算法在迭代过程中较容易陷入局部最优并且无法跳出局部最优,所以通过加入混合蛙跳算法的最差位置改变公式可以使算法在陷入局部最优值时跳出局部最优,从而避免了算法因陷入局部最优无法跳出的问题.图4给出了在测试函数为30维的情况下,分别对单峰函数 $f_4, f_6$ ,多峰函数 $f_7, f_{10}$ 寻

优的结果.

由图4可知,加入混合蛙跳的最差位置改变公式,可以使算法跳出局部最优,图4中Zakharov函数的适应度曲线表示的寻优结果为 $3.68E-28$ ,Quartic函数的寻优结果为 $2.20E-06$ ,Rastrigin函数的寻优结果为0, Penalized 1函数的寻优结果为 $1.97E-02$ .尤其是多峰函数Penalized 1在陷入局部最优值后对比GWO算法的寻优结果, SFL-GWO算法可以有效地跳出局部最优,然后可以去找更加合适的最优值,仿真结果表示了加入混合蛙跳的最差位置改变公式,可克服算法在迭代过程中陷入局部最优的问题.

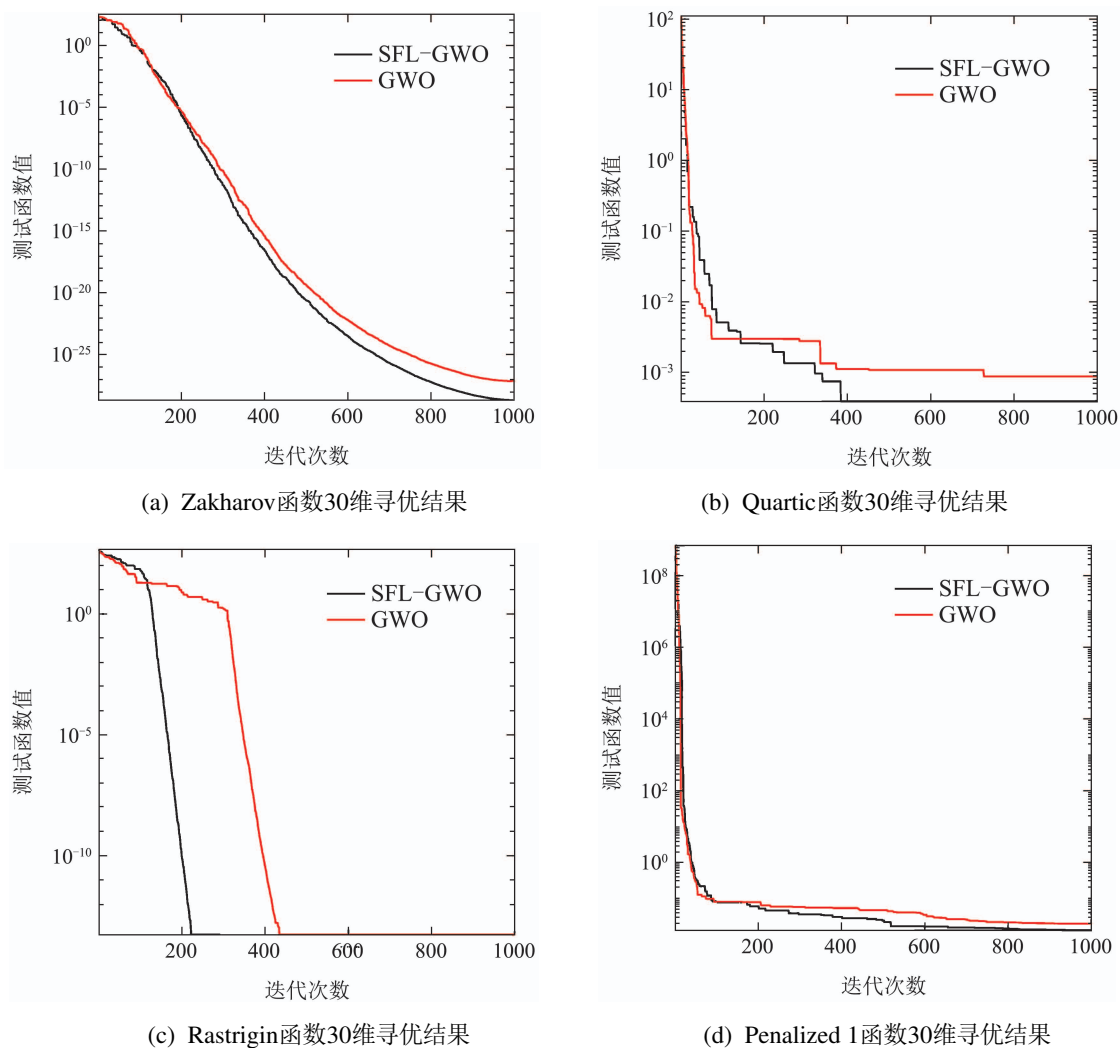


图 4 加入混合蛙跳算法的寻优结果

Fig. 4 Optimization results of shuffled frog leaping algorithm added

综上, 通过对3个部分的单独改进可以看出每部分的改进对算法都有一定的提高, 但是单独改进对于部分测试函数无法找到最优值, 但是对于多数测试函数基本可以满足收敛精度的要求, 最后将对算法的3个部分的改进同时加入到算法中, 可以有效地提高算法的寻优性能以及收敛速度.

### 5.3 不同的算法实验分析

#### 5.3.1 GWO, HGGWO和SFL-GWO算法仿真实验分析

表2中所示, 算法GWO, HGGWO和HFL-GWO在13个30维的函数上进行测试后的结果对比. HG-GWO算法的寻优结果由文献[12]中得到. 通过分析可知, 在1000次的迭代过程中, 改进的算法在全部13个测试函数上都取得了比GWO、HGGWO更好的寻优结果, 对于总共13个测试函数的收敛精度都大大提高了, 尤其是对于 $f_1, f_3, f_5, f_8, f_{10}, f_{11}$ , SFL-GWO算法的搜索结果更是可以都收敛到最优

值0上. 仿真实验可以证明SFL-GWO在求解复杂问题上具有较好的性能指标.

#### 5.3.2 PSO, WOA, GWO和SFL-GWO算法仿真实验分析

在表3中记录了PSO, WOA, GWO和SFL-GWO这4种寻优算法在测试高维复杂函数后的平均值、标准差. 选取PSO, WOA和GWO主要是因为PSO算法是优化领域的一种比较有效地求解方法; WOA算法<sup>[25]</sup>在求解高维复杂函数的过程中有一定的优势; 选用标准GWO算法对比SFL-GWO算法改进前的寻优性能, 在100维、500维、1000维的情况下, 通过对表1的10个函数进行30次独立的运行, 并计算PSO, WOA, GWO及SFL-GWO在30次运行测试后平均值、标准差.

通过分析可知, 对于10个标准测试函数的运行结果分析, 改进的算法的收敛精度要远远高于PSO, WOA和GWO这3种算法. 其中: 在表3中函数的维

度为100时独立寻优运行的测试中, SFL-GWO对于函数 $f_1, f_4, f_5, f_7, f_9$ 可以收敛到最优值0; 在表1中的函数维度为500维时, SFL-GWO算法对函数 $f_1, f_2, f_4, f_5, f_7, f_9$ 的测试结果都收敛到最优值0,  $f_3$ 收敛到最优值-418.9892; 当表3中的函数维度达到500维时, SFL-GWO算法依旧对函数 $f_1, f_4, f_5, f_7, f_9$ 的测试结果还是都可以收敛到最优值0,  $f_3$ 收敛到

最优值-418.9892. 最后, 当表3中的函数维数不断增加到1000维, SFL-GWO算法寻优结果发生微小的改变, 但是收敛精度不但依旧可以满足表1中的要求, 并且对于其中的函数 $f_1, f_4, f_5, f_7, f_9$ 依旧可以收敛到最优值0,  $f_3$ 收敛到最优值-418.9892. 由表3分析可知, SFL-GWO算法在寻优过程中的优势是远远大于算法PSO, WOA和GWO的寻优优势的.

表 2 GWO算法、HGGWO算法以及SFL-GWO算法对13个标准函数的结果对比

Table 2 Comparison of results of 13 standard functions by algorithms GWO, HGGWO and SFL-GWO

函数	GWO算法		HGGWO算法		SFL-GWO算法	
	平均值	标准差	平均值	标准差	平均值	标准差
$f_1$	1.03E-69	5.08E-69	9.56E-118	9.46E-118	<b>0</b>	<b>0</b>
$f_2$	5.97E-41	8.36E-41	2.07E-64	3.49E-64	<b>9.82E-229</b>	<b>0</b>
$f_3$	1.48E-18	8.03E-18	2.63E-30	3.88E-30	<b>0</b>	<b>0</b>
$f_4$	1.83E-17	3.28E-17	2.54E-35	4.85E-35	<b>3.01E-271</b>	<b>0</b>
$f_5$	2.67E+01	2.67E+01	2.52E+01	3.35E-01	<b>0</b>	<b>0</b>
$f_6$	3.10E-01	4.00E-01	3.19E-04	4.16E-04	<b>2.58E-05</b>	<b>3.52E-05</b>
$f_7$	4.85E-04	5.71E-04	7.86E-05	8.25E-05	<b>1.71E-04</b>	<b>2.50E-04</b>
$f_8$	-6.31E+03	5.93E+03	-3.12E+03	3.50E+02	<b>-418.9892</b>	<b>0</b>
$f_9$	2.60E-01	1.40E+0	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
$f_{10}$	1.33E-14	1.37E-14	7.99E-15	2.33E-15	<b>8.88E-16</b>	<b>8.88E-16</b>
$f_{11}$	2.20E-03	6.60E-03	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
$f_{12}$	2.60E-02	2.80E-02	4.57E-04	5.77E-04	<b>3.06E-09</b>	<b>1.38E-08</b>
$f_{13}$	3.02E-01	1.15E-02	1.15E-02	2.17E-02	<b>4.88E-10</b>	<b>1.32E-09</b>

表 3 PSO, WOA, GWO和SFL-GWO算法对10个测试函数在不同维度的寻优结果比较

Table 3 Comparison of optimization results of 10 test functions in different dimensions by PSO, WOA, GWO and SFL-GWO algorithms

函数名	维数	PSO算法		WOA算法		GWO算法		SFL-GWO算法	
		平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
Sphere	100	2.55E+02	2.75E+01	4.13E-168	<b>0</b>	2.00E-34	2.64E-34	<b>0</b>	<b>0</b>
	500	2.14E+04	8.59E+02	3.39E-165	<b>0</b>	2.44E-14	2.67E-14	<b>0</b>	<b>0</b>
	1000	7.86E+04	6.20E+03	9.01E-169	<b>0</b>	4.73E-10	4.91E-10	<b>0</b>	<b>0</b>
Schwefel's2.22	100	6.57E+01	1.32E+01	7.77E-109	3.70E-108	7.82E-21	8.93E-21	<b>8.30E-256</b>	<b>0</b>
	500	4.97E+02	1.36E+02	4.78E-103	1.85E-102	5.17E-09	5.30E-09	<b>0</b>	<b>0</b>
	1000	5.19E+03	1.61E+03	6.44E-109	2.33E-108	2.90E-05	4.72E-05	<b>6.85E-268</b>	<b>0</b>
Schwefel's2.26	100	-2.11E+04	2.08E+04	-3.46E+04	3.46E+04	-1.76E+04	1.72E+04	<b>-418.9892</b>	<b>0</b>
	500	-1.03E+05	1.03E+05	-1.73E+05	1.71E+05	-6.40E+04	6.38E+04	<b>-418.9892</b>	<b>0</b>
	1000	-1.44E+05	1.54E+05	-3.51E+05	3.56E+05	-1.09E+05	1.08E+05	<b>-418.9892</b>	<b>0</b>
Zakharov	100	2.48E+03	2.59E+03	1.76E+03	1.97E+03	7.70E-03	1.17E-02	<b>0</b>	<b>0</b>
	500	7.29E+04	7.40E+04	8.02E+03	8.03E+03	2.69E+03	2.71E+03	<b>0</b>	<b>0</b>
	1000	9.71E+03	5.32E+04	1.60E+04	1.62E+04	6.83E+03	6.89E+03	<b>0</b>	<b>0</b>
Powell	100	368.4577	380.3438	3.42E-30	1.87E-29	1.03E-05	1.53E-05	<b>0</b>	<b>0</b>
	500	8.96E+04	8.99E+04	1.60E-91	8.76E-91	1.19E-05	1.57E-05	<b>0</b>	<b>0</b>
	1000	3.81E+05	3.83E+05	1.92E-126	1.05E-125	1.90E-05	2.70E-05	<b>0</b>	<b>0</b>

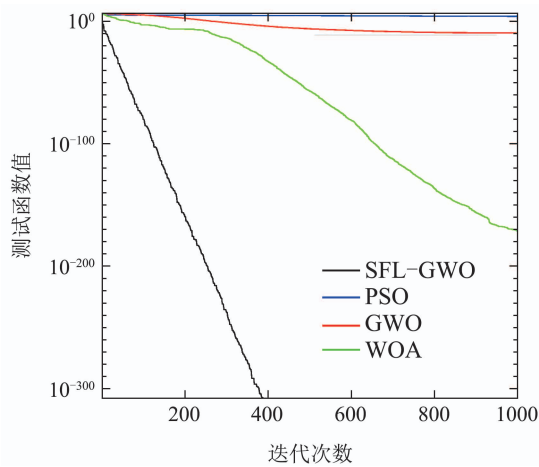


(接上页)

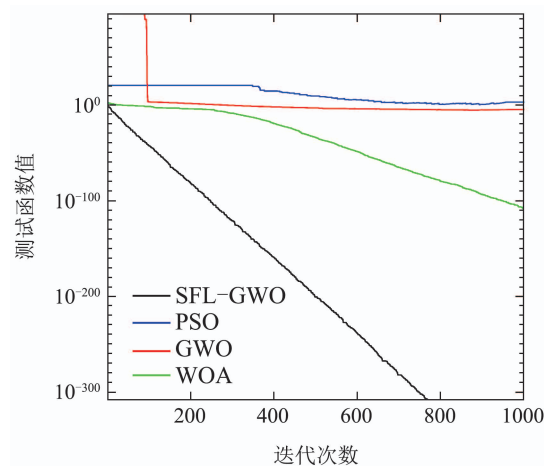
Quartic	100	6.02E+03	5.34E+02	1.40E-03	2.10E-03	1.70E-03	1.80E-03	<b>5.56E-10</b>	<b>1.10E-09</b>
	500	2.43E+04	3.22E+03	8.70E-03	2.80E-02	7.20E-03	7.60E-03	<b>2.72E-05</b>	<b>3.51E-05</b>
	1000	5.04E+05	4.91E+04	4.60E-03	5.10E-03	1.20E-02	1.23E-02	<b>3.46E-05</b>	<b>4.57E-05</b>
Rastrigin	100	3.63E+02	6.38E+01	<b>0</b>	<b>0</b>	7.70E-01	1.17E-02	<b>0</b>	<b>0</b>
	500	4.39E+03	3.24E+02	<b>0</b>	<b>0</b>	2.69E+03	2.71E+03	<b>0</b>	<b>0</b>
	1000	2.77E+04	4.08E+02	<b>0</b>	<b>0</b>	6.83E+03	1.24E+01	<b>0</b>	<b>0</b>
Ackley	100	2.36E+00	4.08E-01	5.50E-15	6.028E-15	6.91E-14	6.93E-14	<b>8.88E-16</b>	<b>8.88E-16</b>
	500	7.34E+00	3.71E-01	3.26E-15	4.29E-15	6.82E-09	7.04E-09	<b>8.88E-16</b>	<b>8.88E-16</b>
	1000	2.47E+01	4.23E+01	4.08E-15	4.69E-15	7.49E-07	7.61E-07	<b>8.88E-16</b>	<b>8.88E-16</b>
Griewank	100	2.44E+02	9.57E+02	<b>0</b>	<b>0</b>	1.10E-03	3.70E-03	<b>0</b>	<b>0</b>
	500	1.37E+02	2.04E+02	<b>0</b>	<b>0</b>	1.30E-03	5.30E-03	<b>0</b>	<b>0</b>
	1000	2.84E+03	1.93E+02	<b>0</b>	<b>0</b>	5.70E-03	1.59E-02	<b>0</b>	<b>0</b>
Penalized 1	100	8.34E-01	7.93E+00	6.30E-03	6.80E-03	1.83E-01	1.86E-02	<b>3.44E-10</b>	<b>5.68E-09</b>
	500	6.22E+03	5.77E+03	1.51E-02	1.15E-02	6.96E-01	6.96E-01	<b>6.36E-10</b>	<b>1.60E-09</b>
	1000	2.57E+06	6.34E+05	1.86E-02	1.92E-02	8.45E-01	8.46E-01	<b>1.92E-09</b>	<b>4.66E-09</b>

总体来说,表3的函数在100维、500维和1000维时,SFL-GWO算法表现出了比PSO,WOA,GWO更好的寻优能力,进一步地证明了SFL-GWO对求解困难问题的高效性.

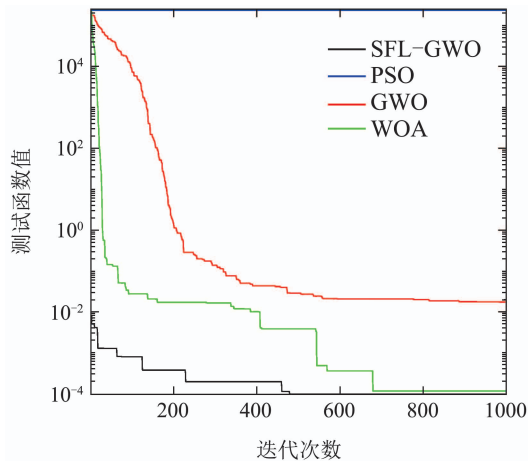
为了更加直观地观察到4种算法收敛的能力,选取单峰函数 $f_1, f_2, f_6, f_7$ 以及多峰函数 $f_8, f_9, f_{10}$ 在1000维的情况下寻优过程中的适应度变化曲线进行对比.适应度变化曲线如图5所示.



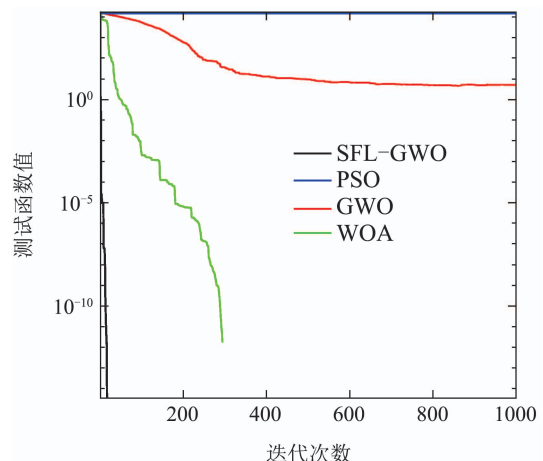
(a) Sphere函数寻优曲线



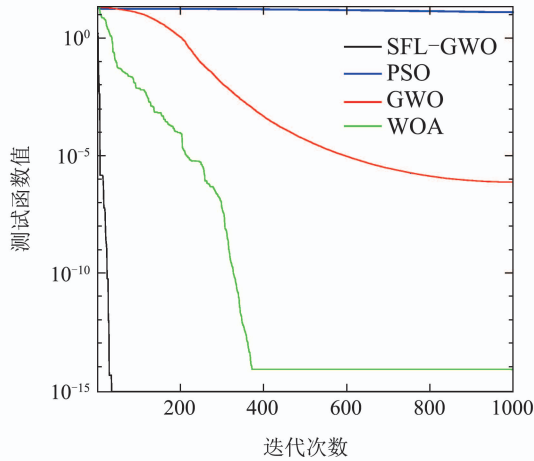
(b) Schwefel's 2.22函数寻优曲线



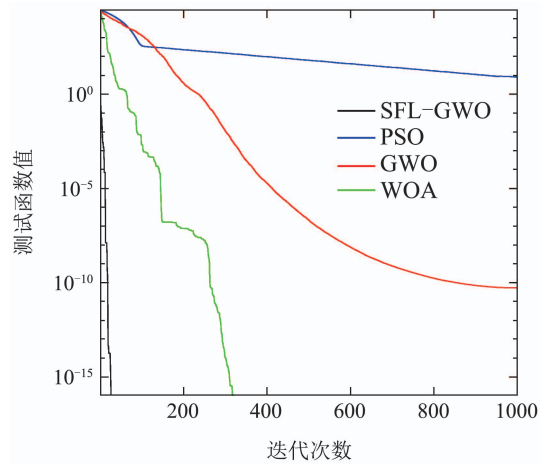
(c) Quartic函数寻优曲线



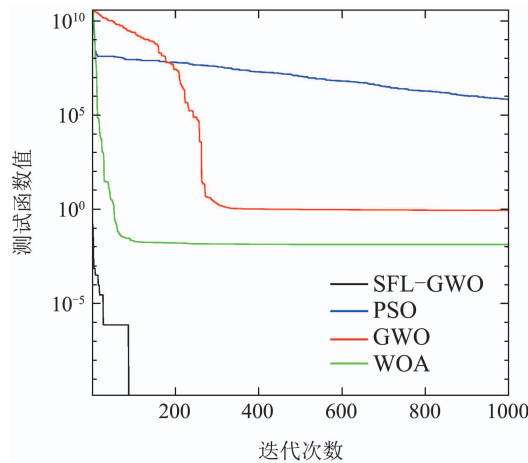
(d) Rastrigin函数寻优曲线



(e) Ackley函数寻优曲线



(f) Griewank函数寻优曲线



(g) Penalized 1函数寻优曲线

图5 PSO, WOA, SFL-GWO及GWO这4种算法的寻优曲线对比

Fig. 5 Optimization curve comparison of PSO, WOA, SFL-GWO and GWO

通过对图5中的7个函数的寻优曲线图可以很直观的看出在SFL-GWO在求解高维复杂函数时,可以在提高收敛速度的同时找到最优值或者求解精度可以满足表1中对函数的收敛精度的要求,对比于PSO, WOA, GWO算法可以明显看出SFL-GWO算法的有效性.

#### 5.4 算法改进策略的分析

SFL-GWO算法采用了3种的改进策略来提高算法的收敛精度和收敛速度,为了验证改进策略的有效性,分别对单独加入某一个策略的算法与GWO算法进行对比分析;当单独加入改进Logistic映射的策略在测试函数上可以找到部分函数的最优值,但是还是有一些函数无法找到最优值;当单独加入改变距离控制参数的策略对所有的测试函数的寻优性能都有提升但是提升效果不是很大;最后,单独加入混合蛙跳最差位置改变公式策略也可以有效的提升了寻优性能.所以3种改进策略都可以提升算法的寻

优性能,但是还是存在部分的不足.

最终,将3种改进的策略同时加入到GWO算法中,通过实验验证分析可以得到结论,SFL-GWO算法在选取的测试函数的寻优结果上都有很大的优势,随着维度的增大,寻优性能并不会有很大的变化.最终可以证明,SFL-GWO算法在求解高维复杂函数上有一定的可靠性.

#### 5.5 算法时间复杂度分析

通过表4分析可得,在对1000维的10个测试函数分别进行30次独立运行并记录平均运行时间的结果分析可以知道,SFL-GWO算法在运行时间上得到了很大地提升,对于多数的单峰以及多峰测试函数,SFL-GWO算法都能很短的时间内找到最优值.综上所述可知,SFL-GWO算法在提高算法寻优精度的同时也提高了算法收敛的时间.但是在对测试函数低维的情况下的运行时间分析,GWO算法的运行时间要少于SFL-GWO算法运行时间.所以,SFL-GWO

算法只适合求解高维复杂函数问题但不太适用于低维函数问题的求解。

表4 时间复杂度  
Table 4 Time complexity

函数	平均运行时间/s	
	GWO	SFL-GWO
Sphere	0.3	1.29
Schwefel's2.22	7.63	1.33
Schwefel's2.26	8.46	1.29
Zakharov	7.51	1.31
Powell	7.62	1.32
Quartic	9.43	1.3
Rastrigin	8.12	1.31
Ackley	8.08	1.33
Griewank	8.52	1.32
Penalized 1	12.37	1.64

## 5.6 与其他算法仿真对比分析

为了进一步地验证SFL-GWO算法的有效性,选取文献[26]中的8种改进算法进行寻优结果的对比及分析. 8种算法分别是: 综合学习粒子群优化算法(comprehensive learning particle swarm optimizer, CLPSO)、正交学习粒子群优化算法(orthogonal

learning particle swarm optimization, OLPSO)、分级粒子群优化算法(hierarchical particle swarm optimizer, HPSO)、具有全局数值优化策略自适应的差分进化算法(differential evolution algorithm with strategy adaptation for global numerical optimization, SaDE)、自适应差分进化算法(adaptive differential evolution, JADE)、自适应控制参数的差分进化算法(selfadapting control parameters in differential evolution, jDE)、受全局最优解指导的人工蜂群算法(gbest-guided artificial bee colony algorithm, GABC)和改进的人工蜂群算法(improved artificial bee colony algorithm, CABC). 从表1的函数中选取单峰函数 $f_1, f_2, f_4, f_6, f_7$ 以及多峰函数 $f_8, f_9, f_{10}$ , 在30维的情况下对算法寻优结果进行分析, 8种改进算法的寻优结果均来自于文献[26].

由表5中可以看出, SFL-GWO算法的寻优性能不仅只是比一些标准的优化算法更好. 在对比其他的改进算法也能体现出很好的寻优性能. 对于测试函数 $f_1, f_4, f_7, f_9$ 都可以达到最优值0; 对于寻优不能达到最优值的函数也能得到比其他算法要更好的结果. 综上, 可以证明SFL-GWO算法在求解高维复杂函数的有效性.

表5 8种改进算法与改进的算法的寻优结果对比

Table 5 Comparison of optimization results of 8 improved algorithms and improved algorithms

函数	HPSO	CLPSO	OLPSO	GABC	SaDE	JADE	jDE	CABC	SFL-GWO
Sphere	2.83E-33	1.58E-12	4.12E-54	1.92E-22	3.28E-20	2.69E-56	1.46E-28	5.41E-35	<b>0</b>
Schwefel's2.22	9.03E-20	2.51E-08	9.85E-30	3.23E-12	3.51E-25	3.18E-25	9.02E-24	1.43E-18	<b>9.82E-229</b>
Rosenbrock	2.39E+01	1.13E+01	2.15E+01	1.15E+00	2.10E+01	3.20E-01	1.30E+01	1.96E-01	<b>0</b>
Quartic	9.82E-02	5.85E-03	1.16E-02	8.94E-02	4.86E-03	6.14E-04	3.35E-03	4.98E-02	<b>2.82E-05</b>
Rastrigin	9.43E+00	9.09E-05	1.07E+00	1.15E-15	2.43E+02	1.33E-01	3.32E-04	<b>0</b>	<b>0</b>
Ackley	7.29E-14	3.66E-07	7.98E-07	2.15E-11	3.81E-06	3.35E-09	2.37E-04	3.07E-14	<b>8.88E-16</b>
Griewank	9.75E-03	9.02E-09	4.83E-03	2.08E-03	2.52E-09	1.57E-08	7.29E-06	4.93E-03	<b>0</b>
Penalized 1	2.71E-29	6.45E-14	1.59E-32	3.07E-24	8.25E-09	1.67E-15	7.03E-08	<b>1.57E-32</b>	1.54E-17

## 6 结论

本文研究了求解高维复杂函数的问题. 通过使用Logistic映射初始化灰狼种群, 改变线性减小的距离控制参数为非线性来提高算法的探索与开发能力, 最后按照设置的改变概率选择ISFLA算法的最差位置改变机制使算法跳出局部最优值. 通过10个高维复杂函数的仿真测试, 与PSO, WOA和GWO 3种基本算法的结果比较, 改进的算法在收敛精度、收敛速度方面得到了很大的提升, 验证了在求解高维函数时上算法SFL-GWO寻优的高效性. 最后, 通过与10种改进算法寻优结果的对比, 结果证

明SFL-GWO算法不论是在求解单峰函数还是多峰函数上, 都表现出比8种改进算法更大的求解优势, 具有更好的全局收敛性.

## 参考文献:

- [1] ZHAI Yinni, SUN Shudong, YANG Hongan, et al. Multi-bottleneck scheduling algorithm for large-scale job shop. *Computer Integrated Manufacturing Systems*, 2011, 17(7): 1486 - 1494. (翟颖妮, 孙树栋, 杨宏安, 等. 大规模作业车间多瓶颈调度算法. 计算机集成制造系统, 2011, 17(7): 1486 - 1494.)
- [2] HUANG Peiqiu, LIU Jianchang, TAN Shubin, et al. Application of the hybrid multi-objective particle swarm optimization algorithm in load distribution of hot finishing mills. *Control Theory & Applica-*

- tions, 2017, 34(1): 93 – 100.  
(黄佩秋, 刘建昌, 谭树彬, 等. 混合多目标粒子群优化算法在热精轧负荷分配优化中的应用. 控制理论与应用, 2017, 34(1): 93 – 100.)
- [3] XU Dongfang, GUO Zhanwei. Intelligent optimization algorithm for high-dimensional and complex functions. *Mathematics in Practice and Theory*, 2016, 46(19): 205 – 211.  
(徐东方, 郭战伟. 求解高维复合体函数的智能优化算法. 数学的实践与认识, 2016, 46(19): 205 – 211.)
- [4] MOHAMMAD NABI OMIDVAR. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 2014, 18(3): 378 – 393.
- [5] HU Chengyu, WANG Bo. Particle swarm optimization with dynamic dimension crossover for high dimensional problems. *Computing Technology and Automation*, 2009, 28(1): 92 – 95  
(胡成玉, 王博. 基于动态维度交叉的粒子群高维函数优化. 计算技术与自动化, 2009, 28(1): 92 – 95.)
- [6] TIAN Jin. Improvement of quantum-behaved particle swarm optimization algorithm for high-dimensional and multi-modal functions. *Control and Decision*, 2016, 31(11): 1967 – 1972.  
(田瑾. 高维多峰函数的量子行为粒子群优化算法改进研究. 控制与决策, 2016, 31(11): 1967 – 1972.)
- [7] YU Wanxia, ZHANG Weicun. Genetic and particle swarm algorithm-based optimization solution for high-dimension complex functions. *Computer Engineering and Applications*, 2007, 43(36): 31 – 33.  
(于万霞, 张维存. 基于遗传粒子群算法的高维复杂函数优化方法. 计算机工程与应用, 2007, 43(36): 31 – 33.)
- [8] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69(3): 46 – 61.
- [9] LONG Wen, WU Tiebin. Improved grey wolf optimization algorithm coordinating the ability of exploration and exploitation. *Control and Decision*, 2017, 32(10): 1749 – 1757.  
(龙文, 伍铁斌. 协调探索和开发能力的改进灰狼优化算法. 控制与决策, 2017, 32(10): 1749 – 1757.)
- [10] LONG Wen, CAI Shaohong, JIAO Jianjun, et al. Hybrid grey wolf optimization algorithm for high-dimensional optimization. *Control and Decision*, 2016, 31(11): 1991 – 1997.  
(龙文, 蔡绍洪, 焦建军, 等. 求解高维优化问题的混合灰狼优化算法. 控制与决策, 2016, 31(11): 1991 – 1997.)
- [11] RODRÍGUEZ L, CASTILLO O, SORIA J. Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic. *Evolutionary Computation. Institute of Electrical and Electronics Engineers*. Vancouver, BC, Canada: IEEE, 2016: 3116 – 3123.
- [12] GU Qinghua, LI Xuexian, LU Caiwu, et al. Hybrid genetic grey wolf algorithm for high dimensional complex function optimization. *Control and Decision*, <https://doi.org/10.13195/j.kzyjc.2018.1253>.  
(顾清华, 李学现, 卢才武, 等. 求解高维复杂函数的遗传-灰狼混合算法. 控制与决策, <https://doi.org/10.13195/j.kzyjc.2018.1253>.)
- [13] LONG Wen, WU Tiebin, TANG Mingzhu, et al. Grey wolf optimizer algorithm based on lens imaging learning strategy. *Acta Automatica Sinica*, <https://doi.org/10.16383/j.aas.c180695>.  
(龙文, 伍铁斌, 唐明珠, 等. 基于透镜成像学习策略的灰狼优化算法. 自动化学报, <https://doi.org/10.16383/j.aas.c180695>.)
- [14] TENG Zhijun, LV Jinling, GUO Liwen, et al. An improved hybrid grey wolf optimization algorithm based on tent mapping. *Journal of Harbin Institute of Technology*, 2018, 50(11): 40 – 49.  
(滕志军, 吕金玲, 郭力文, 等. 一种基于Tent映射的混合灰狼优化的改进算法. 哈尔滨工业大学学报, 2018, 50(11): 40 – 49.)
- [15] JIANG Tianhua. Low-carbon workshop scheduling problem based on grey wolf optimization. *Computer Integrated Manufacturing Systems*, 2018, 24(10): 2428 – 2435.  
(姜天华. 基于灰狼优化算法的低碳车间调度问题. 计算机集成制造系统, 2018, 24(10): 2428 – 2435.)
- [16] LONG Wen, CAI Shaohong, JIAO Jianjun, et al. Hybrid grey wolf optimization algorithm for high-dimensional optimization. *Control and Decision*, 2016, 31(11): 1991 – 1997.  
(龙文, 蔡绍洪, 焦建军, 等. 求解高维优化问题的混合灰狼优化算法. 控制与决策, 2016, 31(11): 1991 – 1997.)
- [17] ZHANG X Q, MING Z F. An optimized grey wolf optimizer based on a mutation operator and eliminating-reconstructing mechanism and its application. *Frontiers of Information Technology & Electronic Engineering*, 2017, 18(11): 1705 – 1719.
- [18] CHEN Zhigang, LIANG Diqing, DENG Xiaohong. Performance analysis and improvement of ILogistic chaotic mapping. *Journal of Electronics & Information Technology*, 2016, 38(6): 1547 – 1551.  
(陈志刚, 梁涤青, 邓小鸿. Logistic混沌映射性能分析与改进. 电子与信息学报, 2016, 38(6): 1547 – 1551.)
- [19] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Water Resour Plan Manage*, 2003, 129(3): 210 – 225.
- [20] PAN Z, MU C L, HU X G, et al. Boundedness of solutions in a chemotaxis system with nonlinear sensitivity and logistic source. *Journal of Mathematical Analysis & Applications*, 2015, 424(1): 509 – 522.
- [21] LONG Wen, CAI Shaohong, JIAO Jianjun, et al. An improved grey wolf optimization algorithm. *Acta Electronica Sinica*, 2019, 47(1): 169 – 175.  
(龙文, 蔡绍洪, 焦建军, 等. 一种改进的灰狼优化算法. 电子学报, 2019, 47(1): 169 – 175.)
- [22] WEI Zhenglei, ZHAO Hui, LI Mudong, et al. A grey wolf optimization algorithm based on nonlinear adjustment strategy of control parameter. *Journal of Air Force Engineering University (Natural Science Edition)*, 2016, 17(3): 68 – 72.  
(魏政磊, 赵辉, 李牧东, 等. 控制参数非线性调整策略的灰狼优化算法. 空军工程大学学报(自然科学版), 2016, 17(3): 68 – 72.)
- [23] DU Jiang, YUAN Zhonghua, WANG Jingqin. Shuffled frog leaping algorithm based on grey prediction theory. *Transactions of China Electrotechnical Society*, 2017, 32(15): 190 – 198.  
(杜江, 袁中华, 王景芹. 一种基于灰预测理论的混合蛙跳算法. 电工技术学报, 2017, 32(15): 190 – 198.)
- [24] ENGELBRECHT A P. *Fundamentals of Computational Swarm Intelligence*. New Jersey: John Wiley & Sons, 2006.
- [25] M IRJALILI S, LEWIS A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, 95(5): 51 – 67.
- [26] GAO W F. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Transactions on Cybernetics*, 2013, 43(3): 1011 – 1024.

#### 作者简介:

**黄晨晨** 硕士研究生, 主要从事群体智能算法优化方向的研究, E-mail: 1305102438@qq.com;

**魏霞** 副教授, 研究生导师, 主要从事工业大数据技术和控制(可再生能源领域)、电力监控系统网络安全防护方向的研究, E-mail: 30462111@qq.com;

**黄德启** 讲师, 从事智能交通方向的研究, E-mail: dqhuang88@qq.com;

**叶家豪** 硕士研究生, 主要从事优化算法、风功率预测等方向的研究, E-mail: 2660343911@qq.com.