

# 强化狼群等级制度的灰狼优化算法

张新明<sup>1,2</sup> 涂强<sup>1</sup> 康强<sup>1</sup> 程金凤<sup>1</sup>

(1. 河南师范大学计算机与信息工程学院, 新乡, 453007; 2. 河南省高校计算智能与数据挖掘工程技术研究中心, 新乡, 453007)

**摘要:** 针对灰狼优化(Grey wolf optimization, GWO)算法在处理复杂优化问题时优化精度不高, 易陷于局部最优等问题, 提出了一种强化狼群等级制度的灰狼优化(GWO based on strengthening the hierarchy of wolves, GWOSH)算法。该算法为灰狼个体设置了跟随狩猎和自主探索两种狩猎模式, 并根据自身等级情况来控制选择狼群的狩猎模式。在跟随狩猎模式中, 灰狼个体以等级高于自身的灰狼的位置信息来指引自己到达最优解区域; 而在自主探索模式中, 灰狼个体会同时审视等级高于自身的灰狼的位置信息和自身位置信息, 并基于这些信息自主判断猎物的位置, 同时两种更新模式都将引入优胜劣汰选择规则来确保种群的狩猎方向。对12个基准测试函数进行优化的结果表明: 与已有的算法相比, GWOSH算法的全局搜索能力更强, 更能有效避免易早熟收敛的问题, 更适用于求解高维的复杂优化问题。

**关键词:** 智能优化算法; 灰狼优化算法; 社会等级制度; 狩猎模式; 复杂优化问题

**中图分类号:** TP181      **文献标志码:** A

## Grey Wolf Optimization Algorithm Based on Strengthening Hierarchy of Wolves

Zhang Xinming<sup>1,2</sup>, Tu Qiang<sup>1</sup>, Kang Qiang<sup>1</sup>, Cheng Jinfeng<sup>1</sup>

(1. College of Computer and Information Engineering, Henan Normal University, Xinxiang, 453007, China; 2. Engineering Technology Research Center for Computing Intelligence & Data Mining, Xinxiang, 453007, China)

**Abstract:** Aiming at the low precision and local optima stagnation of the grey wolf optimization (GWO) algorithm in dealing with complex optimization problems, a grey wolf optimization algorithm based on strengthening the hierarchy of wolves (GWOSH) is proposed. The new algorithm provides two kinds of hunting-modes which are following hunting mode and self-exploration mode for each grey wolf, and each grey wolf chooses its hunting-mode according to the social hierarchy of their own. In the following hunting mode, the grey wolf only depends on the position of higher level wolves to guide itself to search the optimal area. In the self-exploration mode, the individuals will examine the location of the higher level grey wolf and its position at the same time, and judge the position of prey independently based on these information. In the two hunting-modes, a survival of the fittest selection rule is introduced to ensure the evolutionary direction of the population. The optimization results on 12 benchmark functions show that GWOSH has stronger global searching ability and is more effective in the premature convergence avoid-

ance and more suitable for solving high-dimensional complex optimization problems compared with the available algorithms.

**Key words:** intelligent optimization algorithm; grey wolf optimization algorithm; social hierarchy; hunting-modes; complex optimization problems

## 引言

群体智能优化算法源于对自然界中生物进化过程或觅食行为的模拟<sup>[1]</sup>,其中具有代表性的有蚁群优化算法<sup>[2]</sup>,粒子群优化算法<sup>[3]</sup>,细菌觅食算法<sup>[4]</sup>和人工蜂群(Artificial bee colony, ABC)算法<sup>[5]</sup>等。这些算法都具有良好的自组织学习性和高效的搜索机制,在复杂优化问题的求解上有着突出的表现,目前已成为优化领域的研究热点。同时这些算法也各具特色,如蚁群算法采用了正反馈机制完成种群间的信息传递,并通过蚂蚁个体的共同协作以实现最终收敛于最优路径的目的,但算法局部搜索能力差,收敛速度慢且优化性能受参数设置影响较大;粒子群优化算法机制简单,具有较高的运行效率,且受问题维数影响较小,但算法易陷入局部最优,搜索性能也过于依赖参数设置;细菌觅食优化算法具有较好的局部搜索能力,但算法结构复杂,运行效率偏低;ABC算法鲁棒性强,适用于各种复杂连续优化问题,且不敏感于参数和初值的选择,但算法搜索精度不高且随机性过强。依据无免费午餐定理,没有一种算法完全适用于所有问题的求解,研究者们便通过不断提出新的算法和对已有算法性能的改进来达到寻求更好解决方案的目的。

灰狼优化算法(Grey wolf optimization, GWO)是 Mirjalili 于 2014 年提出的一种新颖的群体智能优化算法<sup>[6]</sup>。该算法模拟了自然界中灰狼家族的社会等级制度和群体狩猎行为,算法中等级最低的  $\omega$  狼在等级较高的  $\alpha$  狼,  $\beta$  狼和  $\delta$  狼的带领下,通过跟踪猎物、包围猎物和攻击猎物来完成狩猎,属于一种有导向的随机性启发式算法。GWO 算法因具有不过分依赖参数设置、全局搜索能力强和易于实现等优点,目前已成功应用于作业车间调度<sup>[7]</sup>、工程优化<sup>[8]</sup>、支持向量机分类<sup>[9]</sup>和经济调度指派<sup>[10]</sup>等领域,有效地解决了多种优化问题。但 GWO 在解决复杂优化问题时仍然容易过早陷入局部极值,即出现早熟收敛的现象<sup>[11]</sup>。

为了克服上述不足,基于灰狼家族的社会等级制度对于狼群狩猎影响的深入分析,提出了一种强化狼群等级制度的灰狼优化(GWO based on strengthening the hierarchy of wolves, GWOSH)算法。该算法为灰狼个体设置了两种狩猎模式:跟随狩猎模式和自主探索模式,并根据决策因子来选择狼群的狩猎模式。对 12 个基准测试函数的仿真结果表明,与原始的 GWO<sup>[6]</sup>、双模飞行的粒子群优化(Particle swarm optimization with double-flight modes, DMPSO)算法<sup>[12]</sup>、正弦差分进化算法(Sinusoidal differential evolution, SinDE)<sup>[13]</sup>和 ABC<sup>[5]</sup>算法相比,GWOSH 能较好地寻得全局最优解,且比原始的 GWO 算法具有更快的收敛速度。

## 1 原始灰狼优化算法

灰狼优化算法是一种基于种群的元启发式优化算法,它模拟了自然界中灰狼家族的社会等级制度和其群体的狩猎行为。

### 1.1 灰狼家族的社会等级制度

自然界中的灰狼是一种以群居生活为主的食肉动物,它们之间有着严格的等级管理制度,图 1 给出了灰狼种群等级制度的示意图。灰狼家族中的狼按照社会地位从高到低可以划分为 4 类,分别是: $\alpha$  狼、 $\beta$  狼、 $\delta$  狼和  $\omega$  狼。如图 1 所示, $\alpha$  狼是处于社会顶层的狼,也叫“头狼”,它是狼群的最高统治者和管

理者,负责决定狼群狩猎的时间、地点及战术。位于第2层的是 $\beta$ 狼,相当于狼群中的副首领,负责接替头狼继续领带狼群。位于最底层的是 $\omega$ 狼, $\omega$ 狼总是要必须服从其他所有地位高的狼。而属于第3层的是 $\delta$ 狼, $\delta$ 狼必须服从 $\alpha$ 狼和 $\beta$ 狼,但可以统治 $\omega$ 狼,主要负责侦查、放哨以及看护工作<sup>[14]</sup>。

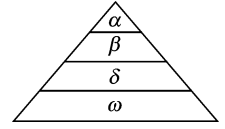


图1 灰狼种群等级制度示意图

Fig.1 Sketch map of the hierarchy of grey wolf

## 1.2 GWO 算法描述

在GWO算法中,首先构建灰狼的社会等级制度模型。将种群中适应度值最优的,次优的和第三优的解分别对应 $\alpha$ 狼, $\beta$ 狼和 $\delta$ 狼,而剩余的被称为 $\omega$ 狼。由 $\alpha$ 狼, $\beta$ 狼和 $\delta$ 狼来负责引导, $\omega$ 狼则跟着 $\alpha$ 狼, $\beta$ 狼和 $\delta$ 狼通过搜寻猎物、包围猎物和攻击猎物来完成狩猎(优化)。

(1) 包围猎物。灰狼包围猎物的数学模型为<sup>[6]</sup>

$$D = |C \cdot X_p - X^t| \quad (1)$$

$$X^{t+1} = X_p - A \cdot D \quad (2)$$

式(1)定义了灰狼与猎物之间的距离,而式(2)则定义了灰狼的最终位置。其中, $t$ 表示当前的迭代次数, $X_p$ 表示猎物的位置, $X$ 表示灰狼的位置, $A$ 和 $C$ 为参数向量,其计算公式为<sup>[6]</sup>

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

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

式中: $a$ 为整个迭代过程中一个从2到0的线性递减参数, $r_1$ 和 $r_2$ 为 $[0,1]$ 之间的随机向量。

(2) 狩猎。为了模拟狼群的集体狩猎行为,总假设 $\alpha$ 狼, $\beta$ 狼和 $\delta$ 狼对于猎物的潜在位置有更好的了解。因此,在每次迭代过程中,保存当前获得的适应度值最优( $\alpha$ 狼)、次优( $\beta$ 狼)及第三优( $\delta$ 狼)的3头狼的位置信息,然后狼群根据这3个最优解的位置信息综合判断出个体向猎物移动的方向并更新自己的位置,其更新公式为<sup>[6]</sup>

$$D_\alpha = |C_1 \cdot X'_\alpha - X^t| \quad (5)$$

$$D_\beta = |C_2 \cdot X'_\beta - X^t| \quad (6)$$

$$D_\delta = |C_3 \cdot X'_\delta - X^t| \quad (7)$$

$$X_1 = X'_\alpha - A_1 \cdot D_\alpha \quad (8)$$

$$X_2 = X'_\beta - A_2 \cdot D_\beta \quad (9)$$

$$X_3 = X'_\delta - A_3 \cdot D_\delta \quad (10)$$

$$X^{t+1} = (X_1 + X_2 + X_3) / 3 \quad (11)$$

(3) 攻击猎物(开发)。为了构建灰狼攻击猎物的数学模型,根据式(3), $A$ 为区间 $[-2a, 2a]$ 上的随机向量,其中 $a$ 在迭代过程中是从2到0的一个线性递减参数,当 $|A|$ 的值在 $[-1, 1]$ 内时,灰狼的下一个位置将会在它现在位置和猎物位置之间的任意位置,表示向猎物发起进攻。

(4) 搜寻猎物(探索)。灰狼主要根据 $\alpha$ 狼, $\beta$ 狼和 $\delta$ 狼的位置信息来搜寻猎物,狼群先分散开来寻找猎物然后聚集在一起来攻击猎物。为了构建分散模型,利用一个随机值大于1或小于-1的 $A$ 来迫使灰狼背离猎物去探索更有希望的搜索空间,以便实现算法的全局搜索。GWO算法中的另外一个探索向量是 $C$ ,从式(4)中可以看出 $C$ 是 $[0, 2]$ 上的随机值,它能随机增加( $|C| > 1$ )或减轻( $|C| < 1$ )灰狼靠近猎物时的难易程度,使GWO算法在整个优化过程中显示出更多的随机行为,以便提高算法的全局探索能力。

## 2 强化狼群等级制度的灰狼优化算法

### 2.1 GWOSH 算法的基本思想

从GWO算法可知,它的狩猎模型是先由 $\alpha$ 狼, $\beta$ 狼和 $\delta$ 狼共同负责对猎物的位置进行评估定位,然

后其余个体以此为标准计算自身与猎物之间的距离,并完成对猎物的全方位靠近、包围和攻击等行为,最终完成狩猎。在这个过程中 $\alpha$ 狼、 $\beta$ 狼、 $\delta$ 狼及 $\omega$ 狼的等级特性体现的并不是很明显,而 $\alpha$ 狼、 $\beta$ 狼和 $\delta$ 狼的位置信息对于其他个体的位置更新又起着绝对的引导作用,容易致使整个狼群过早聚集于群体当前最优位置的某一邻域内。另外,探索能力还受制于两个探索参数 $A$ 和 $C$ 的较小取值范围,从而导致算法易陷于局部最优。

而由第1节可知,灰狼家族有着严格的社会等级管理制度,不同等级的灰狼享有不同的权利和社会分工,灰狼所属的等级越高,对猎物的情况就有更好的了解,自主能动性也越强,且这种等级制度在狼群实现团体高效捕杀猎物的过程中发挥着至关重要的作用。因此,本文提出一种强化狼群等级制度的灰狼优化算法,以期提高其性能。

## 2.2 GWOSH 算法

设狼群的搜索空间为一个 $N \times D$ 的欧式空间,其中 $N$ 为灰狼数量, $D$ 为待寻优的变量数。设第 $i$ 只灰狼 $t$ 时刻的位置为 $\mathbf{X}_i(t) = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$ ,当前找到具有最优适应值的 $\alpha$ 狼位置为 $\mathbf{X}_\alpha^t = (x_{\alpha 1}^t, x_{\alpha 2}^t, \dots, x_{\alpha D}^t)$ ,当前次优适应值的 $\beta$ 狼位置为 $\mathbf{X}_\beta^t = (x_{\beta 1}^t, x_{\beta 2}^t, \dots, x_{\beta D}^t)$ ,当前第三适应值的 $\delta$ 狼位置为 $\mathbf{X}_\delta^t = (x_{\delta 1}^t, x_{\delta 2}^t, \dots, x_{\delta D}^t)$ 。

为了强化等级制度对于狼群狩猎的影响,为每个灰狼个体设计两种狩猎模式:跟随狩猎模式和自主探索模式。为了设计算法模型,作出如下假设:(1)每头狼采用跟随狩猎模式和自主探索模式;(2)每头狼依据一定的规则来确定其狩猎模式。跟随狩猎模式中的灰狼个体的位置更新原理类似于原始GWO算法,只是仅以等级高于自身的灰狼的位置信息来指引自己到达最优解区域;而在自主探索模式中,灰狼个体会同时审视等级高于自身灰狼的位置信息和自身位置信息,并以这些信息为基础采用随机变向的方式来更新自己的位置,同时两种更新模式都将引入优胜劣汰选择规则来确保种群的狩猎方向。

每只灰狼会根据自身所属等级情况来确定其狩猎模式。GWOSH算法先将候选解按适应度值的优劣划分为4个等级,适应度值最优者为 $\alpha$ 狼,其等级为1;适应度值次优者为 $\beta$ 狼,其等级为2;适应度值第三优者为 $\delta$ 狼,其等级为3;余下的候选解为 $\omega$ 狼,其等级为4,记第 $i$ 头狼 $t$ 时刻的等级为 $S_i(t)$ ,则 $S_i(t)=1$ 表示第 $i$ 头灰狼 $t$ 时刻在群体中所属等级为1, $S_i(t)=2$ 表示第 $i$ 头灰狼 $t$ 时刻在群体中所属等级为2,依次类推。然后,用决策因子 $DF_i(t)$ 来标记每一头灰狼个体的等级情况,其计算公式为

$$DF_i(t) = \frac{L - S_i}{L - 1} \quad (12)$$

式中: $L$ 为狼群所分等级总数,即 $L=4$ 。从式(12)可看出,对于等级为1的 $\alpha$ 狼而言,其决策因子 $DF_i(t)=1$ ,等级为2的 $\beta$ 狼的决策因子 $DF_i(t)=0.67$ ,等级为3的 $\delta$ 狼的决策因子 $DF_i(t)=0.33$ ,等级为4的 $\omega$ 狼的决策因子为0。

在每一次的迭代过程中,对于每一个候选解都会先生成一个 $[0,1]$ 间的随机数 $r_0$ ,如果 $r_0 \leq DF_i(t)$ ,则灰狼 $i$ 在 $t$ 时刻就会采用自主探索模式,否则采用跟随狩猎模式。显然对于 $\alpha$ 狼因其决策因子为1,而 $r_0$ 恒小于或等于1,所以 $\alpha$ 狼有且仅有一种狩猎模式即自主探索狩猎模式,对于 $\omega$ 狼因其决策因子为0,而 $r_0$ 大于或等于0,所以 $\omega$ 狼仅能采用跟随狩猎模式。

对于 $\alpha$ 狼,其位置更新公式为

$$\begin{cases} \mathbf{X}_{id}^{t+1} = \mathbf{X}_{\alpha d}^t & d \neq k \\ \mathbf{X}_{id}^{t+1} = \mathbf{X}_{\alpha d}^t + 2 \cdot a \cdot r_3 \cdot (\mathbf{X}_{md}^t - \mathbf{X}_{nd}^t) & d = k \end{cases} \quad (13)$$

式中: $k$ 为 $\{1,2,\dots,D\}$ 中的一个随机数; $m$ 和 $n$ 为区间 $[1,N]$ 内与 $i$ 不等的随机整数,且两两互不相等; $a$ 为整个迭代过程中从2到0的一个线性递减参数, $r_3$ 为区间 $[0,1]$ 中的一个随机数, $\mathbf{X}_\alpha$ 表示具有最优适应值的 $\alpha$ 狼的位置。

对于 $\beta$ 狼,按式(14)进行位置更新,即有

$$\begin{cases} \mathbf{X}_{id}^{t+1} = \mathbf{X}_{1d} & r_0 > 0.67 \\ \mathbf{X}_{id}^{t+1} = (\mathbf{X}_{\alpha k}^t + \mathbf{X}_{\beta k}^t)/2 & r_0 \leq 0.67 \end{cases} \quad (14)$$

式中: $k$ 为 $\{1, 2, \dots, D\}$ 中的一个随机数; $\mathbf{X}_1$ 来自于式(8); $\mathbf{X}_\alpha$ 和 $\mathbf{X}_\beta$ 分别表示 $\alpha$ 狼和 $\beta$ 狼的位置。

对于 $\delta$ 狼,其按式(15)进行位置更新,即有

$$\begin{cases} \mathbf{X}_{id}^{t+1} = (\mathbf{X}_{1d} + \mathbf{X}_{2d})/2 & r_0 > 0.33 \\ \mathbf{X}_{id}^{t+1} = (\mathbf{X}_{\alpha k}^t + \mathbf{X}_{\beta k}^t + \mathbf{X}_{\delta k}^t)/3 & r_0 \leq 0.33 \end{cases} \quad (15)$$

式中: $k$ 为 $\{1, 2, \dots, D\}$ 中的一个随机数; $\mathbf{X}_1$ 和 $\mathbf{X}_2$ 分别来自于式(8)和式(9); $\mathbf{X}_\alpha$ 、 $\mathbf{X}_\beta$ 和 $\mathbf{X}_\delta$ 分别表示 $\alpha$ 狼、 $\beta$ 狼和 $\delta$ 狼的位置。

对于 $\omega$ 狼,按式(16)进行位置更新,即有

$$\mathbf{X}_{id}^{t+1} = (\mathbf{X}_{1d} + \mathbf{X}_{2d} + \mathbf{X}_{3d})/3 \quad (16)$$

式中: $\mathbf{X}_1$ 、 $\mathbf{X}_2$ 和 $\mathbf{X}_3$ 分别来自于式(8-10)。

为了确保种群的进化方向,GWOSH算法还引入了优胜劣汰选择规则来保留每一代的最优解,即如果新个体的适应度值 $f(\text{new})$ 优于上一代该个体的适应度值 $f(\text{old})$ ,则用新个体的位置替代上一代个体的位置,否则保留上一代个体的位置,该规则可表示为

$$x_{id}^{t+1} = \begin{cases} x_{id}^{t+\text{new}} & f(\text{old}) \leq f(\text{new}) \\ x_{id}^{t+\text{old}} & \text{其他} \end{cases} \quad (17)$$

GWOSH算法流程具体如下:

- (1) 参数初始化。设置种群个数为 $N$ ,最大迭代次数为 $\text{Maxgen}$ ;
- (2) 根据变量的上下界来随机初始化灰狼个体的位置 $\mathbf{X}_i$ ,其中, $i=1, 2, \dots, N$ ;
- (3) 计算狼群中每一头狼的适应度值;
- (4) 按适应度值从优到差的顺序对狼群进行排序,并分别保存当前获得的具有最优、次优和第三优适应度值的灰狼的位置信息 $\mathbf{X}_\alpha$ 、 $\mathbf{X}_\beta$ 和 $\mathbf{X}_\delta$ 。同时,用 $S_i(t)$ 来标记每个灰狼个体的所属等级,适应度值最优者 $S_i(t)=1$ ;次优者 $S_i(t)=2$ ;第三优者 $S_i(t)=3$ ,剩下的灰狼个体 $S_i(t)=4$ ;
- (5) 对狼群位置进行更新。 $S_i(t)=1, 2, 3$ 和 $4$ 的灰狼分别根据式(13-16)进行位置更新;
- (6) 计算狼群中每一头狼的适应度值;
- (7) 依据优胜劣汰选择规则,按式(17)确定是否接受该新位置;
- (8) 更新参数 $a$ 、 $A$ 和 $C$ ;
- (9) 判断是否到达最大迭代次数 $\text{Maxgen}$ ,若满足则停止并返回 $\mathbf{X}_\alpha$ 的值作为最终得到的最优解,否则转到(4)。

## 3 仿真实验及结果分析

### 3.1 GWOSH算法

为了验证本文提出的GWOSH算法的性能,将其用于高维复杂函数优化计算,同时将计算结果与原始GWO、DMPSO、SinDE及ABC算法进行对比分析,其中的DMPSO和SinDE都是对其标准算法的一个有效改进,而ABC算法也是近年来提出较为新颖的优化算法,据相关文献报道其性能优于GA、PSO等传统优化算法。选取12个常用的典型高维函数进行算法性能测试分析,其表达式和全局最优解等情况如表1所示,其中12个函数的维数都为60; $f_1 \sim f_4$ 为高维单峰函数,主要用来考察算法的寻优精度; $f_5 \sim f_8$ 为复杂的非线性多峰函数,由于局部极值点较多,算法较难寻找到全局最优解,因此主要用来检验算法规避早熟的能力和全局搜索的能力;同时, $f_9 \sim f_{12}$ 是对基本测试函数进行了平移操作,使得函数具有更复杂的结构,主要用来检验算法对于未知空间的搜索能力<sup>[15-17]</sup>。本文所有实验均在CPU主频3.4GHz的Intel Core(TM)i7-3770和内存8GB的PC机上进行,操作系统为Microsoft Windows 7,编程工具为MATLAB R2014a。

表 1 测试函数  
Tab. 1 Test functions

Name	Function	Dim	Range	F <sub>min</sub>
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	60	$[-100, 100]$	0
Schwefel2.22	$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	60	$[-10, 10]$	0
Rosenbrock	$f_3(x) = \sum_{i=1}^n [(x_i - 1)^2 + 100(x_{i+1} + x_i^2)^2]$	60	$[-10, 10]$	0
Step	$f_4(x) = \sum_{i=1}^n ( x_i + 0.5 )^2$	60	$[-100, 100]$	0
Ackley	$f_5(x) = 20 + e - 20 \exp[-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}] - \exp[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)]$	60	$[-32, 32]$	0
Griewank	$f_6(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$	60	$[-600, 600]$	0
FN13	$f_7(x) = -\frac{1}{n} \sum_{i=1}^n \exp(-2 \ln 2 (\frac{x_i - 0.1}{0.8})^2) \times [\sin^6(5\pi x_i) + 0.1 \cos^2(500\pi x_i)]$	60	$[0, 1]$	-1.1
Rastrigin	$f_8(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	60	$[-5.12, 5.12]$	0
Shifted Sphere	$f_9(z) = \sum_{i=1}^n z_i^2 - 450, z = x - o$	60	$[-100, 100]$	-450
Shifted Schwefel's Problem 2.21	$f_{10}(z) = \max_{i=1}^n \{ z_i \} - 450, z = x - o$	60	$[-100, 100]$	-450
Shifted Griewank	$f_{11}(z) = 1 + \sum_{i=1}^n \frac{z_i^2}{4000} - \prod_{i=1}^n \cos(\frac{z_i}{\sqrt{i}}) - 180, z = x - o$	60	$[-600, 600]$	-180
Shifted Ackley	$f_{12}(z) = 20 + e - 20 \exp(-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}) - \exp[\frac{1}{n} \sum_{i=1}^n \cos(2\pi z_i)] - 140, z = x - o$	60	$[-32, 32]$	-140

### 3.2 仿真实验结果

为了确保测试的公平性,在本实验中,GWOSH与原始GWO、DMPSO、SinDE以及ABC算法的种群规模和最大迭代次数设置保持一致,即每种算法种群规模统一设置为40,最大迭代次数都设置为1000,5种算法的其他参数设置如表2所示。这些参数都设置为在保证能使各自算法的寻优效果最佳,且能够稳定收敛时所取得的值。

**实验1** 优化性能比较。选取5个算法在12个典型函数上进行30次寻优计算的最优值(Best)、最差值(Worst)、平均值(Mean)和均方差(Std)

作为各算法寻优性能的评价指标。表3-5分别列出了5个算法在12个典型函数上进行30次寻优计算的所得结果,最优者用黑体表示,图2显示了5种算法在12个典型函数上寻优过程中平均最优适应度值随迭代次数的变化情况,其中图2(a-f),图2(h)的纵坐标均取以10为底的对数值。

表 2 优化算法的参数设置

Tab. 2 Parameter setting of the optimization algorithms

优化算法	参数设置
DMPSO	学习因子 $C_1$ 和 $C_2$ 均为 2, 惯性权重 $w=0.628$
SinDE	缩放因子 $F=0.25$
ABC	控制参数 $\text{Limit}=2400$
GWO	参数 $a$ 初始值为 2
GWOSH	参数 $a$ 初始值为 2

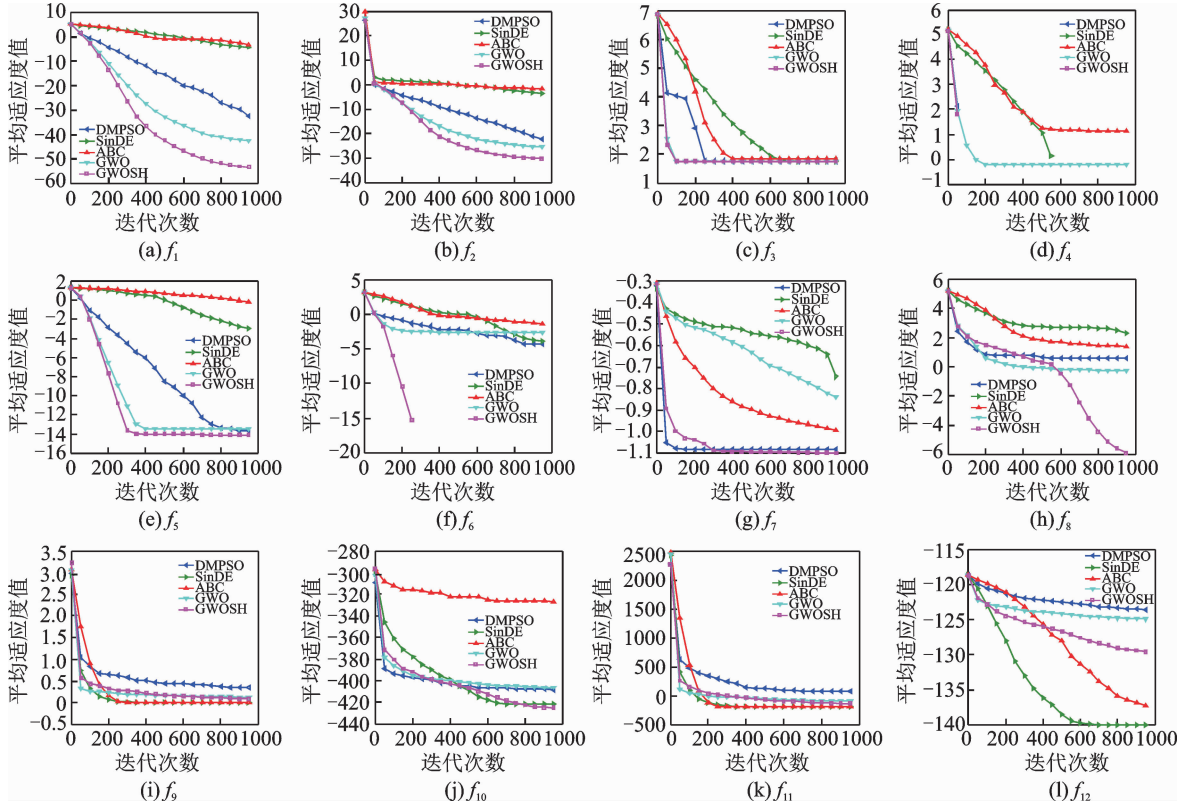


图 2 5 种算法在  $f_1 \sim f_{12}$  上的收敛曲线

Fig. 2 Convergence curves of five algorithms on  $f_1 \sim f_{12}$

表 3 列出了 5 种算法对于 4 个单峰函数的寻优结果,由表 3 可知,对于  $f_1 \sim f_4$  函数,GWOSH 相对于其他 4 种算法,在搜索的最优值、最差值、平均值和均方差上均取得了最好的优化结果,尤其是在  $f_4$  函数上获得了全局最优解,虽然 DMPSO 也在  $f_4$  函数上获得了全局最优解,但根据图 2 中  $f_4$  函数的收敛图可以看出 GWOSH 具有更快的收敛速度。结合表 3 和图 2 可以看出,GWOSH 的收敛速度和搜索精度大幅优于其他 4 种算法。这是因为高等级灰狼对于猎物的位置有更好的了解,使得 GWOSH 中的灰狼个体仅跟随更高等级的灰狼进行寻优,避免了由于自身位置受低等级灰狼的影响所造成的解的退化,同时优胜劣汰选择规则的引入也确保了狼群的进化方向,大幅提高了搜索效率及寻优精度。

表 4 列出了 5 种算法对于 4 个复杂多峰函数的寻优结果,由表 4 可以看出,对于  $f_5 \sim f_8$  函数,GWOSH 在 5 种算法中,均获得了最好的解,尤其对于  $f_6$  函数获得了全局最优解。由图 2 也可以看出 GWOSH 算法对于  $f_5, f_6, f_7$  和  $f_8$  函数具有更快的全局收敛速度,表明狼群在经过若干次迭代运算后仍具备从局部最优中跳出的能力,即使是在多模态空间,由于自由探索模式中的灰狼是以更高等级灰狼及自身所处的位置信息为参照并通过随机变向的方式来更新自身位置,增加了狼群多样性,迫使狼群不断探索新区域,从而提高了算法规避陷入局部最优的能力。

表 5 列出了 5 种算法对于 4 个平移函数的寻优结果,因为平移后的基准测试函数有着更加复杂的搜索空间,对于所有优化算法都是一个很大的挑战,结合表 5 和图 2 可以看出,SinDE 算法对于  $f_9, f_{11}$  和  $f_{12}$  函数有着最好的寻优效果,其次是 ABC 算法。虽然 GWOSH 在  $f_9, f_{11}$  和  $f_{12}$  函数上的寻优效果差于 SinDE 和 ABC 算法,但其在  $f_{12}$  函数上取得了最好的解,其最优值、最差值、平均值和均方差都优于其他 4 种对比算法,且 GWOSH 算法在 4 个函数上得到的最优值、最差值、平均值和均方差均要明显优

表 3 5 种算法在单峰函数上的测试结果对比

Tab. 3 Test result comparison of five algorithms on unimodal benchmark functions					
Fun	Algorithm	Best	Worst	Mean	Std
$f_1$	DMP SO	9.295 1e-40	1.107 2e-30	3.766 0e-32	2.020 1e-31
	SinDE	2.703 5e-06	9.760 3e-04	6.690 5e-05	1.778 2e-04
	ABC	1.402 4e-04	0.013 2	0.001 6	0.002 5
	GWO	1.392 7e-44	1.212 7e-42	2.605 5e-43	3.216 5e-43
	GWOSH	<b>9.185 6e-51</b>	<b>4.491 3e-62</b>	<b>5.572 3e-52</b>	<b>1.921 3e-51</b>
$f_2$	DMP SO	5.279 2e-26	5.806 7e-22	6.773 8e-23	1.288 8e-22
	SinDE	8.259 3e-05	6.056 7e-04	2.227 9e-04	1.149 7e-04
	ABC	0.007 6	0.058 4	0.021 2	0.012 8
	GWO	8.304 3e-27	4.428 2e-25	8.927 4e-26	8.511 6e-26
	GWOSH	<b>2.885 7e-34</b>	<b>3.301 3e-29</b>	<b>2.168 5e-30</b>	<b>6.187 5e-30</b>
$f_3$	DMP SO	<b>1.856 2e-16</b>	371.446 9	78.336 9	133.115 3
	SinDE	53.892 9	119.271 3	63.534 2	17.679 0
	ABC	12.624 1	180.316 3	74.943 1	45.748 5
	GWO	55.196 9	58.548 0	57.106 1	0.910 6
	GWOSH	54.972 1	<b>56.909 0</b>	<b>55.899 4</b>	<b>0.521 0</b>
$f_4$	DMP SO	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	SinDE	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	ABC	2	28	12	6.756 9
	GWO	<b>0</b>	4	0.600 0	0.932 2
	GWOSH	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

表 4 5 种算法在多峰函数上的测试结果对比

Tab. 4 Test result comparison of five algorithms on multimodal benchmark functions					
Fun	Algorithm	Best	Worst	Mean	Std
$f_5$	DMP SO	<b>2.664 5e-15</b>	7.371 9e-14	2.208 6e-14	1.699 3e-14
	SinDE	2.487 9e-04	0.004 0	0.001 4	0.001 2
	ABC	0.0401	1.252 5	0.519 9	0.319 0
	GWO	2.753 4e-14	4.174 4e-14	3.463 9e-14	4.376 1e-15
	GWOSH	<b>2.664 5e-15</b>	<b>1.332 3e-14</b>	<b>6.454 1e-15</b>	<b>1.597 9e-15</b>
$f_6$	DMP SO	<b>0</b>	0.001	3.408 6e-05	1.859 7e-04
	SinDE	9.016 6e-07	0.007 4	3.205 7e-04	0.001 3
	ABC	1.674 8e-04	0.093 9	0.029 3	0.028 3
	GWO	<b>0</b>	0.035 7	0.003 4	0.008 5
	GWOSH	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
$f_7$	DMP SO	<b>-1.100 0</b>	-1.004 3	-1.086 8	0.031 6
	SinDE	-0.912 0	-0.839 7	-0.880 6	0.015 0
	ABC	-1.029 0	-0.980 2	-1.003 4	0.009 6
	GWO	-0.952 0	-0.782 7	-0.860 3	0.041 3
	GWOSH	<b>-1.100 0</b>	<b>-1.008 6</b>	<b>-1.093 7</b>	<b>0.003 1</b>
$f_8$	DMP SO	<b>0</b>	59.697 5	5.969 8	18.215 4
	SinDE	83.355 8	190.621 7	129.198 2	20.323 9
	ABC	9.241 2	32.057 3	21.362 3	6.324 2
	GWO	<b>0</b>	16.580 9	1.029 6	3.586 6
	GWOSH	<b>0</b>	<b>1.065 8e-13</b>	<b>1.065 8e-14</b>	<b>3.252 1e-14</b>



表 5 5 种算法在平移函数上的测试结果对比

Tab. 5 Test result comparison of five algorithms on shifted functions

Fun	Algorithm	Best	Worst	Mean	Std
$f_9$	DMP SO	2.841 6e+04	4.174 4e+04	3.484 1e+04	3.195 2e+03
	SinDE	<b>-450.000 0</b>	<b>-449.997 7</b>	<b>-449.999 8</b>	<b>4.539 3e-04</b>
	ABC	-449.999 3	-449.924 1	-449.984 0	0.017 7
	GWO	4.220 1e+03	1.785 8e+04	9.418 7e+03	4.212 0e+03
	GWOSH	2.862 3e+03	1.511 3e+04	7.704 8e+03	3.851 8e+03
$f_{10}$	DMP SO	-412.963 8	-399.806 7	-406.411 9	4.729 1
	SinDE	-432.544 3	-406.212 2	-420.238 1	6.610 3
	ABC	-338.762 4	-312.781 5	-321.980 2	6.057 7
	GWO	-412.950 3	-401.594 8	-404.854 0	3.286 5
	GWOSH	<b>-439.264 2</b>	<b>-420.092 3</b>	<b>-425.626 6</b>	<b>2.780 1</b>
$f_{11}$	DMP SO	36.170 60	113.503 2	65.892 5	20.212 7
	SinDE	<b>-180.000 0</b>	<b>-179.992 1</b>	<b>-179.999 2</b>	<b>0.002 0</b>
	ABC	-179.992 8	-179.479 5	-179.864 6	0.125 9
	GWO	-147.430 2	16.763 80	-93.023 8	42.748 2
	GWOSH	-165.351 0	-66.563 8	-129.291 8	26.718 7
$f_{12}$	DMP SO	-124.961 6	-122.707 6	-123.795 5	0.509 0
	SinDE	<b>-139.999 7</b>	<b>-139.985 6</b>	<b>-139.997 9</b>	<b>0.002 7</b>
	ABC	-138.329 5	-136.759 9	-137.606 5	0.393 9
	GWO	-131.862 9	-123.521 0	-126.792 4	2.029 5
	GWOSH	-132.566 7	-125.068 1	-129.645 0	1.939 3

于原始 GWO 算法,这表明 GWOSH 算法较原始 GWO 对于未知的搜索空间有着更好的寻优能力。

**实验 2 运行时间比较。**表 6 列出了 5 种算法对于 12 个优化函数独立运行 30 次的平均运行时间,其单位为 s。因为本文实验采用最大迭代次数作为算法终止条件,且每种算法种群规模相同,即最大函数评价次数相同,函数计算耗时相同,故表 6 的时间差异可反映各算法自身结构的差异。由表 6 可以看出,相对于其他 4 种算法,GWOSH 的运行时间最少,表明其具有较快的寻优速度。因为该算法虽然有

表 6 5 种算法运行时间的对比

Tab. 6 Comparison of running time for five algorithms

Fun	DMP SO	SinDE	ABC	GWO	GWOSH
$f_1$	1.001 0	0.647 19	0.793 55	0.819 56	<b>0.528 26</b>
$f_2$	1.104 0	0.738 51	0.939 49	0.898 17	<b>0.547 88</b>
$f_3$	1.161 3	0.812 77	0.960 06	1.003 20	<b>0.490 95</b>
$f_4$	1.057 7	0.684 30	0.822 58	0.873 26	<b>0.496 65</b>
$f_5$	1.240 8	0.915 58	1.081 90	1.016 80	<b>0.591 35</b>
$f_6$	1.721 0	1.427 00	1.596 10	1.448 50	<b>0.593 64</b>
$f_7$	1.863 9	1.719 80	1.786 90	1.773 50	<b>0.845 99</b>
$f_8$	1.133 3	0.827 63	0.968 38	0.887 13	<b>0.582 94</b>
$f_9$	1.731 6	1.365 90	1.461 40	1.457 20	<b>0.557 68</b>
$f_{10}$	1.752 0	1.387 80	1.519 60	1.416 10	<b>0.541 44</b>
$f_{11}$	2.439 4	2.053 30	2.221 10	2.145 20	<b>0.671 97</b>
$f_{12}$	1.968 0	1.602 80	1.785 90	1.680 40	<b>0.672 55</b>
Average	1.514 5	1.181 88	1.328 08	1.284 92	<b>0.593 44</b>

两种搜索模式,但二者交替进行,无需额外的计算,而且在算法设计模式上,DMP SO, SinDE 和 ABC 在计算适应度函数值时均采用串行计算的方式,而 GWOSH 的设计可采用大规模的并行运算模式,从而大幅提高了算法的运行速度。而 GWO 算法直接采用该算法原创者提供的代码,未对代码进行优化处理和采用串行计算模式,故其耗时比 GWOSH 多。综上所述,GWOSH 算法不管在单模态函数、多模态函数,还是平移的复杂函数上,与原始 GWO 算法相比都具有更优秀的优化性能,单模态函数和多模态函数的优化性能超过 DMP SO、SinDE 和 ABC 算法,所以本文算法的改进是有效和可行的。

## 4 结束语

GWO 算法是最近提出的群智能优化算法,有许多方面值得研究。本文针对原始 GWO 算法在优化过程中存在的易陷入局部最优等缺点,基于自然界中灰狼的等级制度对于狼群狩猎影响的深入分析,提出了一种强化狼群等级制度的灰狼优化算法。该算法中的灰狼个体具有跟随狩猎模式和自主探索模式两种狩猎模式,可根据决策因子来控制选择狼群的狩猎模式。这两种狩猎模式既能充分体现狼群中高等级灰狼对低等级灰狼的引领作用,又能在充分挖掘种群位置信息的基础上发挥个体的自主能动性,提高种群的多样性,避免陷入局部最优。仿真实验结果表明:本文提出的改进策略是有效的,改进算法的全局搜索能力更强,更适用于求解高维的复杂优化问题。

## 参考文献:

- [1] Sundar S, Singh A. A swarm intelligence approach to the quadratic minimum spanning tree problem [J]. *Information Science*, 2010, 180(17): 3182-3191.
- [2] Zhang X M, Song W, Feng W. Improved ant colony algorithm for parameter estimation on the BISQ mode [J]. *Inverse Problems in Science and Engineering*, 2015, 23(6): 997-1010.
- [3] 王养廷, 闫文忠. 分组信息共享的量子粒子群优化算法的改进 [J]. *数据采集与处理*, 2013, 28(3): 363-370.  
Wang Yangting, Yan Wenzhong. Improvement of grouped information sharing particle swarm optimization based on quantum [J]. *Journal of Data Acquisition and Processing*, 2013, 28(3): 363-370.
- [4] 张新明, 尹欣欣, 冯梦清. 动态高斯变异和随机变异融合的自适应细菌觅食优化算法 [J]. *计算机科学*, 2015, 42(6): 101-106.  
Zhang Xinming, Yin Xinxin, Feng Mengqing. Adaptive bacterial foraging optimization algorithm based on dynamic Gaussian mutation and random one for high dimensional function [J]. *Computer Science*, 2015, 42(6): 101-106.
- [5] 张新明, 魏峰, 牛丽平, 等. 混合排名映射概率和混沌搜索的 ABC 算法 [J]. *计算机科学*, 2014, 41(2): 102-106.  
Zhang Xinming, Wei Feng, Niu Liping, et al. Artificial bee colony algorithm based on hybrid rank mapping probability and chaotic search [J]. *Computer Science*, 2014, 41(2): 102-106.
- [6] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. *Advances in Engineering Software*, 2014, 69(3): 46-61.
- [7] KoMaki G M, Kayvanfar V. Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time [J]. *Journal of Computational Science*, 2015, 8: 109-120.
- [8] Sulaiman M H, Mustaffa Z, Mohamed M R, et al. Using the gray wolf optimizer for solving optimal reactive power dispatch problem [J]. *Applied Soft Computing*, 2015, 32(C): 286-292.
- [9] Mirjalili S. How effective is the grey wolf optimizer in training multi-layer perceptrons [J]. *Applied Intelligence*, 2015, 43(1): 150-161.
- [10] Song H M, Sulaiman M H, Mohamed M R. An application of grey wolf optimizer for solving combined economic emission dispatch problems [J]. *International Review on Modelling and Simulations*, 2014, 7(5): 838-844.
- [11] Zhu A J, Xu C P, Zhi L, et al. Hybridizing grey wolf optimization with differential evolution for global optimization and test

scheduling for 3D stacked SoC [J]. Journal of Systems Engineering and Electronics, 2015, 26(2): 317-328.

- [12] 李景洋, 王勇, 李春雷. 采用双模飞行的粒子群优化算法 [J]. 模式识别与人工智能, 2014, 27(6): 533-539.  
Li Jingyang, Wang Yong, Li Chunlei. Particle swarm optimization algorithm with double-flight modes [J]. Pattern Recognition and Artificial Intelligence, 2014, 27(6): 533-539.
- [13] Draa A, Bouzoubia S, Boukhalfa I. A sinusoidal differential evolution algorithm for numerical optimization [J]. Applied Soft Computing, 2015, 27(27): 99-126.
- [14] Pilot M, Branicki W, Jędrzejewski W, et al. Phylogeographic history of grey wolves in Europe [J]. BMC Evolutionary Biology, 2010, 10(1):104.
- [15] Kiran M S, Findik O. A directed artificial bee colony algorithm [J]. Applied Soft Computing, 2015, 26(C):454-462.
- [16] Hakli H, Uguz H. A novel swarm optimization algorithm with Levy flight [J]. Applied Soft Computing, 2014, 23(5): 333-345.
- [17] Tanweer M R, Suresh S, Sundararajan N. Self regulating particle swarm optimization algorithm [J]. Information Sciences, 2015, 294: 182-202.

#### 作者简介:



**张新明** (1963-), 男, 教授, 研究方向: 模式识别、数字图像处理 and 智能优化算法等, E-mail: xinmingzhang@126.com。



**涂强** (1995-), 男, 硕士研究生, 研究方向: 数字图像处理和智能优化算法。



**康强** (1989-), 男, 硕士研究生, 研究方向: 数字图像处理和智能优化算法。



**程金凤** (1990-), 女, 硕士研究生, 研究方向: 数字图像处理。

