

基于混合扰动的非线性灰狼优化算法

李征南, 秦江涛

(上海理工大学管理学院, 上海 200093)

✉ 486453073@qq.com; qinjiangtao_usst@126.com



摘要:针对传统灰狼优化算法(GWO)求解精度低、收敛速度慢和易陷入局部最优的缺点,提出一种基于混合扰动的非线性灰狼优化算法。首先,该算法采用一种新型非线性控制参数策略和基于余弦变换的惯性权重策略在探索和开发阶段取得平衡;其次,为了增加种群在算法后期的多样性,采用 t -分布概率扰动策略;最后,对灰狼最优个体进行扰动,增强算法跳出局部最优的能力。为了验证改进 GWO 算法的性能,在 10 个基准函数上对其进行测试,通过与 4 种基本智能算法、4 种单一改进策略的 GWO 算法及 4 种其他改进算法进行比较,结果表明:改进 GWO 算法在 90% 的测试函数上优于 4 种基本智能算法,在 50% 的测试函数上优于 4 种单一改进策略的 GWO 算法,在 50% 的测试函数上优于 4 种其他改进算法。

关键词:灰狼优化算法;控制参数; t -分布

中图分类号:TP301.6 **文献标识码:**A

Nonlinear Grey Wolf Optimization Algorithm Based on Hybrid Perturbation

LI Zhengnan, QIN Jiangtao

(Business School, University of Shanghai for Science & Technology, Shanghai 200093, China)

✉ 486453073@qq.com; qinjiangtao_usst@126.com

Abstract: Aiming at the shortcomings of traditional Grey Wolf Optimization algorithm (GWO), such as low solution precision, slow convergence speed and being easy to fall into local optimum, this paper proposes a nonlinear Grey Wolf Optimization algorithm based on hybrid disturbance. Firstly, a new nonlinear control parameter strategy and an inertial weight strategy based on cosine transform are used to achieve a balance between exploration and development. Secondly, in order to increase the diversity of the population in the later stage of the algorithm, a t -distribution probability perturbation strategy is used. Finally, the Grey Wolf optimal individual is perturbed to enhance the ability of the algorithm to jump out of the local optimum. In order to verify the performance of the improved GWO algorithm, it is tested on 10 benchmark functions and compared with four basic intelligent algorithms, four single improved strategy GWO algorithms and four other improved algorithms, the results show that the improved GWO algorithm is superior to the four basic intelligent algorithms on 90% of the test functions. It is better than four single improvement strategies on 50% test function and better than four other improved algorithms on 50% test functions.

Keywords: Grey Wolf Optimization algorithm; control parameters; t -distribution

0 引言 (Introduction)

灰狼优化算法(GWO)是 2014 年由 MIRJALILI 等^[1]提出

的一种种群智能优化算法。该算法因模型简单、参数少、性能好而受到一些学者的追捧,被广泛应用于函数优化、参数寻优和

故障诊断等领域^[2-4]。尽管传统的灰狼优化算法在应用中表现出较佳的性能,但在面对多峰函数优化时,仍然存在依赖初始种群分布、容易陷入局部最优、难以有效协调开发和探索能力的缺点。目前,国内外学者从不同角度对灰狼优化算法进行了改进。王正通等^[5]引入新型动态扰动因子策略,以确保算法的寻优精度,并将翻筋斗觅食策略融入 GWO 算法中,提高灰狼种群的多样性,使算法不易陷入局部最优。徐明等^[6]结合非线性调整策略、个体记忆策略及小孔成像学习策略,提出一种多策略 GWO 算法,并对特征选择问题进行求解。为了平衡探索和开发阶段的能力,MA 等^[7]提出了一种基于天鹰优化器(Aquila Optimizer, AO)的改进 GWO 算法。

为有效改进 GWO 求解精度低、易陷入局部最优的不足,本文探索了 4 种不同的策略改进传统 GWO 算法。改进后的非线性控制参数和基于余弦变换的惯性权重,用于 GWO 从探索到开发的和谐过渡。在每次的寻优过程中对灰狼个体采取 t -分布概率扰动策略和最优个体扰动策略,增加灰狼的探索能力和提高灰狼个体跳出局部最优的能力,并采用贪婪策略决定是否更新最优灰狼。将所有策略都合并到 GWO 中,提出一种新的算法,称作改进的灰狼优化算法(Improved Grey Wolf Optimization Algorithm, IGWO)。仿真结果表明,IGWO 在 10 个基准函数上具有更快的收敛速度和更高的收敛精度。

1 灰狼优化算法(Grey Wolf Optimization algorithm)

在自然界中,灰狼是一种群居犬科动物,在生物圈食物链的顶端有着严格的等级制度。如图 1 所示,灰狼等级按照金字塔从上往下依次划分,即 α 、 β 、 δ 和 ω 。 α 对其他灰狼有绝对的控制权,主要管理其他等级的狼。 β 对除 α 外的其他灰狼具有绝对优势,并辅助 α 进行决策。狼 δ 服从 α 和 β 的命令,对 ω 有绝对的控制权。灰狼 ω 是服从其他级别命令的狼。在 GWO 算法中,一般通过计算灰狼的目标函数值,将适应度最好(即函数值最低)的 3 只灰狼分别记为 α 、 β 和 δ ,灰狼 ω 在它们的引导下向猎物位置前进。

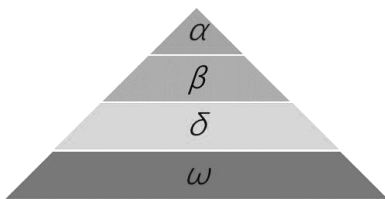


图 1 灰狼等级制度

Fig. 1 The gray wolf hierarchy

1.1 包围猎物

灰狼包围猎物的数学模型如下所示:

$$D = |CX_p(t) - X(t)| \quad (1)$$

$$X(t+1) = X_p(t) - AD \quad (2)$$

$$A = 2ar_1 - a \quad (3)$$

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

其中, D 表示灰狼与猎物之间的距离; $X_p(t)$ 为猎物的位置; 灰狼当前位置是 $X(t)$; A 和 C 是系数向量; a 是控制参数, 随着迭代次数的不断增加, 从 2 线性减少到 0。 r_1 和 r_2 是 $[0, 1]$ 中的随机向量。

1.2 狩猎

在狩猎阶段, 灰狼 ω 会根据其他三只领导狼的位置更新其位置, 其数学模型如下所示:

$$\begin{cases} D_\alpha = |C_1 X_\alpha - X| \\ D_\beta = |C_2 X_\beta - X| \\ D_\delta = |C_3 X_\delta - X| \end{cases} \quad (5)$$

$$\begin{cases} X_1 = X_\alpha - A_1 D_\alpha \\ X_2 = X_\beta - A_2 D_\beta \\ X_3 = X_\delta - A_3 D_\delta \end{cases} \quad (6)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (7)$$

公式(5)至公式(7)中, 灰狼 α 、 β 和 δ 的位置分别为 X_α 、 X_β 、 X_δ , D_α 、 D_β 和 D_δ 分别代表当前候选灰狼与灰狼 α 、 β 、 δ 之间的距离。 X_1 、 X_2 和 X_3 分别代表灰狼 ω 分别朝灰狼 α 、 β 和 δ 前进的方向和步长, $X(t+1)$ 是灰狼 ω 的最终位置。

1.3 攻击猎物

当猎物停止移动时, 灰狼会根据 α 、 β 和 δ 的位置来攻击猎物。由上文可知, a 的值从 2 开始逐渐线性减小到 0, 所以 a 的取值范围也不断缩小, 其对应的 A 值也会在区间 $[-a, a]$ 内发生变化。当 A 的取值在 $[-1, 1]$ 范围内时, 灰狼发动攻击, 对应于局部搜索; 反之, 灰狼就会扩大搜索范围, 转向探索。

2 改进的灰狼优化算法(Improved Grey Wolf Optimization algorithm)

2.1 非线性控制参数和余弦变换的权值因子

在群智能优化算法中, 整个进化过程可分为探索(全局搜索)和开发(局部搜索)两个阶段。较强的局部搜索能力, 会加快算法收敛的速度; 而较强的全局搜索能力, 会增强算法跳出局部最优的能力。因此, 如何有效协调探索和开发能力, 是算法获得高搜索性能的关键。从上文可知, A 值的大小决定了 GWO 是进行全局搜索还是局部搜索, 而 A 值和控制参数 a 呈线性正相关。但是, GWO 算法是非线性搜索, 控制参数 a 的线性递减策略不能完全反映实际的优化搜索过程。

从文献[8]得知, 进行全局搜索与局部搜索的比例为 7 : 3 时, 算法搜索性能更佳。文献[9]比较了多种不同函数形式的控制参数, 发现余弦函数和二次函数的控制参数更有助于算法性能的提升。因此, 本文提出一种搜索比例为 7 : 3 的二次函数形式的非线性控制参数:

$$a = -\frac{40}{21}\left(\frac{t}{T}\right)^2 - \frac{2}{21}\left(\frac{t}{T}\right) + 2 \quad (8)$$

公式(8)中, t 为当前迭代次数; T 为最大迭代次数, 其值设为 500。由图 2 可知, 控制参数 a 随着迭代次数的不断增加而呈非线性动态变化, 在算法迭代初期衰减较为缓慢, 搜索步长相对较大, 利于算法进行全局搜索; 迭代后期衰减加快, 步长

减小,有利于提升算法的搜索精度,能够有效协调算法的全局搜索能力和局部搜索能力。

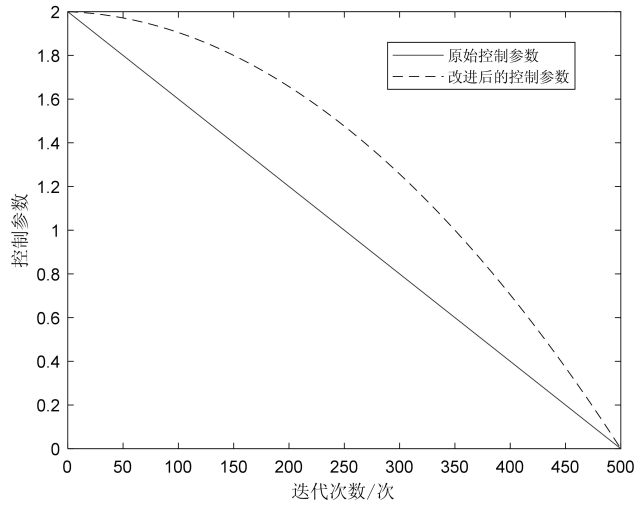


图2 控制参数对比

Fig. 2 Comparison of control parameter

虽然非线性控制参数对 GWO 的性能提升有一定的作用,但是效果有限,在算法的全局搜索和局部搜索之间难以平衡。考虑到粒子群优化算法(Particle Swarm Optimization, PSO)中惯性权重对算法的有效改进,为了更好地协调算法的全局搜索和局部搜索能力,本文提出一种基于余弦变换的自适应权值策略,如公式(9)所示。权值因子 w 和迭代次数 t 具有正相关的关系,即随着 t 的增加而呈非线性增加的趋势。算法初期,惯性权重较小,局部寻优能力得到了增强,从而提高了算法的寻优精度;算法中后期,灰狼种群聚集度较高,此时惯性权重随迭代次数的增加而变大,有利于灰狼种群发挥全局寻优能力,提高算法的寻优速度。公式(10)是改进后的灰狼位置更新公式。

$$w = 0.6 \times \cos\left[\frac{\pi}{2} \left(1 - \frac{t}{T}\right)\right] + 10^{-6} \quad (9)$$

$$\mathbf{X}(t+1) = w \times \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3} \quad (10)$$

2.2 t -分布概率扰动

t -分布又称作学生分布,含有自由度参数 n ,其概率密度函数如下:

$$f(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n \times \pi} \times \Gamma(\frac{n}{2})} \times \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}, -\infty < x < +\infty \quad (11)$$

其中, Γ 是 Gamma 函数。

本文提出一种基于最优个体的 t -分布概率扰动策略,利用最优灰狼个体对其他个体进行扰动后,产生了新的随机种群,帮助算法摆脱局部最优。扰动后的个体就会分布在最优灰狼附近,帮助最优灰狼个体寻找到最佳位置,不仅增强了算法的开采能力,还帮助最优个体跳出局部最优点。设置 t -分布概率扰动策略中变异概率 $p = 0.01 \times (2 - a)$,其中 a 是非线性控制

参数。针对每只灰狼,都生成一个 $[0, 1]$ 的随机数,随机数小于变异概率的灰狼为满足变异条件的灰狼, t -分布扰动公式如下:

$$u = 1 - \cos\left[0.5 \times \pi \times \left(1 - \frac{t}{T}\right)\right] \quad (12)$$

$$\mathbf{X}'_i = u \times \text{rand} \times (\mathbf{X}_{\text{best}} - \mathbf{X}_i) + \text{trnd}(t) \quad (13)$$

其中, $\mathbf{X}'_i$ 为第 i 只经过 t -分布概率扰动策略扰动后灰狼个体的位置, $\mathbf{X}_{\text{best}}$ 为最优灰狼个体的位置, $\mathbf{X}_i$ 是第 i 只灰狼个体的位置, $\text{trnd}(t)$ 是自由度参数为迭代次数的 t -分布随机数。 u 是取值 $[0, 1]$ 范围内的系数,随着迭代次数的增加, u 不断变小,最优灰狼对灰狼个体的影响不断减小且变异概率 p 不断变大,以此增加算法变异的概率,在迭代后期,会帮助算法跳出局部最优。

如图3所示, t -分布把柯西分布和高斯分布二者的特点结合。迭代初期,会呈现出柯西分布的特点,丰富了种群的多样性;迭代中后期,自由度参数的取值较大, t -分布近似于高斯分布,增强了算法的局部开发能力。

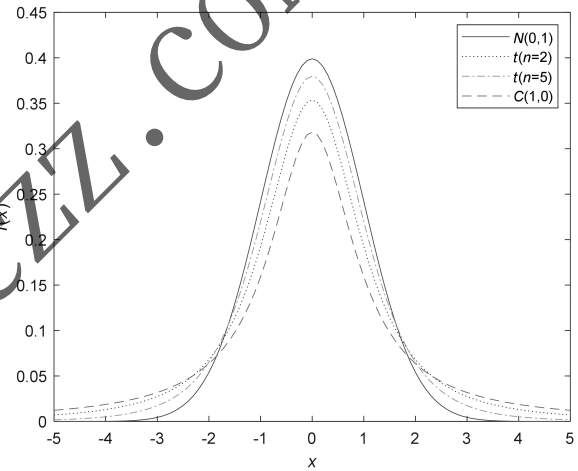


图3 高斯分布、 t -分布和柯西分布函数分布图

Fig. 3 Gaussian distribution, t -distribution and Cauchy distribution function distribution

2.3 最优位置扰动策略

GWO 算法在迭代后期,灰狼种群多样性大大降低,易于陷入局部最优的情况,因此要对最优位置进行扰动,增加对周围空间的搜索。本文结合 t -分布对最优灰狼个体进行扰动,增强算法摆脱局部最优解的能力,保证算法的精度,具体公式如下:

$$\mathbf{X}_{\text{new}} = \mathbf{X}_{\text{best}} \times (0.5 + 0.5 \times \text{trnd}(t)) \quad (14)$$

为了保证经过 t -分布扰动后的灰狼位置优于原目标位置,在更新操作后引入贪婪策略,通过比较新、旧目标位置的适应度值后再决定是否更新目标位置。贪婪策略如下:

$$\mathbf{X}_{\text{best}} = \begin{cases} \mathbf{X}_{\text{new}}, & f(\mathbf{X}_{\text{new}}) \leq f(\mathbf{X}_{\text{best}}) \\ \mathbf{X}_{\text{best}}, & f(\mathbf{X}_{\text{new}}) > f(\mathbf{X}_{\text{best}}) \end{cases} \quad (15)$$

2.4 算法流程

改进灰狼优化算法的伪代码如下:

初始化算法参数:种群规模 n 、最大迭代次数 T 、空间维度 dim ;
随机初始化灰狼种群;

计算灰狼种群中各个体适应度值并排序。适应度值前 3 位分别记为 α, β, δ ;

for $t=1$ to T do

 for $i=1$ to n do

 for $j=1$ to dim do

 使用公式(1)至公式(6)计算灰狼 ω 分别朝狼 α, β, δ 和前

 进的步长和方向;

 使用公式(8)至公式(10)更新灰狼位置;

 end

 end

 for $i=1$ to n do

$p=0.01*(2-a)$

 if $rand < p$ do

 使用公式(12)、公式(13)更新灰狼位置;

 end

 end

计算灰狼适应度值,更新狼 α, β 和 δ 的位置,然后使用公式(14)和公式(15)对最优灰狼的位置进行扰动更新;

end

3 实验仿真与结果分析(Experimental simulation and result analysis)

本实验使用 MATLAB R2019b 进行仿真实验。操作环境为 64 位 Windows 10 操作系统, Intel(R) Core(TM) I5—6300HQ CPU @ 3.0 GHz。为了验证本文改进的灰狼优化算法具有更好的搜索性能和求解精度,实验选用了文献[1]中 10 个标准测试函数进行验证。测试函数信息如表 1 所示,其中函数 $F1 \sim F7$ 为单峰函数,只有一个最小值,所以用来测试算法的开发性能。多峰函数 $F8 \sim F10$, 具有一个全局最优值和多个局部极值,可以用来评估算法的探索能力。

表 1 基准测试函数

Tab. 1 Benchmark function

名称	函数表达式	自变量范围	最优值
$F1$	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
$F2$	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0
$F3$	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]$	0
$F4$	$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]$	0
$F5$	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0
$F6$	$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]$	0
$F7$	$f_7 = \sum_{i=1}^n ix_i^4 + random[0, 1]$	$[-1, 28, 1, 28]$	0
$F8$	$f_8(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5, 12, 5, 12]$	0

续表

名称	函数表达式	自变量范围	最优值
$F9$	$f_9(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	0
$F10$	$f_{10}(x) = \frac{1}{4\,000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0

3.1 与原始 GWO 及其他几种算法对比

将 IGWO 与灰狼优化算法 (GWO)^[1]、粒子群优化算法 (PSO)^[10]、鲸鱼优化算法 (WOA)^[11] 和海鸥优化算法 (SOA)^[12] 进行对比,算法参数设置相同:种群大小为 30 个,空间维度 $dim=30$,最大迭代次数设置为 500 次。同时,为客观评价算法寻优性能的好坏,在上述参数设置条件下,各算法独立运行 30 次,记录各算法的平均值和标准差。其中,平均值越小,表明算法寻优精度越高;标准差越小,算法越稳定。测试结果如表 2 所示,其中加粗字体为最优测试结果。

表 2 各优化算法在 30 维下的寻优对比基准测试函数

Tab. 2 Comparison of optimization results of each optimization algorithm in 30 dimensions

函数	评价	GWO	PSO	WOA	SOA	IGWO
$F1$	mean	2.23E-27	1.47E+01	1.27E-74	7.99E-12	0
	std	2.68E-27	1.88E+00	4.73E-74	1.33E-11	0
$F2$	mean	9.28E-17	1.68E+01	6.50E-50	1.89E-08	0
	std	5.67E-17	1.27E+00	3.28E-49	1.47E-08	0
$F3$	mean	2.83E-05	9.50E+01	4.39E+04	5.22E-05	0
	std	7.43E-05	2.81E+01	4.06E+04	9.16E-05	0
$F4$	mean	8.68E-07	1.63E+00	4.79E+01	1.71E-02	4.17E-317
	std	1.07E-06	9.97E-02	2.45E+01	5.65E-02	0
$F5$	mean	2.71E+01	3.39E+03	2.79E+01	2.83E+01	1.03E-01
	std	6.86E-01	7.38E+02	4.89E-01	5.59E-01	2.11E-01
$F6$	mean	6.54E-01	1.56E+01	4.18E-01	3.24E+00	3.00E-03
	std	3.42E-01	2.18E+00	2.44E-01	5.16E-01	6.70E-03
$F7$	mean	2.10E-03	1.06E+02	3.80E-03	3.20E-03	5.97E-05
	std	1.20E-03	1.76E+01	3.70E-03	2.70E-03	7.33E-05
$F8$	mean	2.67E+00	2.33E+02	0	3.98E+00	0
	std	2.60E+00	2.00E+01	0	1.40E+01	0
$F9$	mean	1.02E-13	4.33E+00	4.09E-15	1.99E+01	8.88E-16
	std	1.92E-14	1.85E-01	2.85E-15	1.80E-03	0
$F10$	mean	4.10E-03	6.41E-01	7.40E-03	1.96E-02	0
	std	8.30E-03	7.97E-02	4.03E-02	3.32E-02	0

表 2 展示了 IGWO 和各种基本算法在函数 $F1 \sim F10$ 上的运行结果。由表 2 得知,在维度 $dim = 30$ 时,IGWO 算法在 5 个测试函数 $F1, F2, F3, F8, F10$ 上能找到最优值。而对于函数 $F4 \sim F7, F9$, IGWO 算法虽然未能达到理论最优值,但是寻优结果非常逼近最优值。同时,对比函数 $F1 \sim F10$ 的标准差(std),发现 IGWO 的标准差(除 $F8$ 外)均小于其他 4 种基本算法的标准差,仅在函数 $F8$ 上测试所得标准差与 WOA 算法相同,并小于其他 3 种算法,这表明 IGWO 算法收敛稳定性更强,算法的鲁棒性更好。综合来看,IGWO 算法的寻优能力优于其他 4 种算法。

3.2 与不同改进策略的对比

将 IGWO 与仅改进控制参数的灰狼优化算法(GWO1)、仅采用自适应惯性权重的灰狼优化算法(GWO2)、仅采用 t -分布扰动策略改进的灰狼优化算法(GWO3)和仅实行最优位置扰动策略的灰狼优化算法(GWO4)进行比较,参数设置与本文“3.1”相同,以验证 IGWO 的优越性。表 3 展示了各种改进 GWO 算法在测试函数上的均值与标准差,其中加粗字体为最优的测试结果。此外,为更清晰地展示各单一策略对 GWO 算法的影响,图 4 给出了各种算法在函数 $F3 \sim F6, F8 \sim F9$ 上的收敛曲线图。

表 3 各改进 GWO 算法寻优结果对比

Tab. 3 Comparison of optimization results of each improved GWO algorithm

函数	评价	GWO	GWO1	GWO2	GWO3	GWO4	IGWO
F1	mean	2.23E-27	3.03E-36	0	1.03E-26	0	0
	std	2.68E-27	4.59E-36	0	1.41E-26	0	0
F2	mean	9.28E-17	1.21E-21	3.71E-276	6.27E-16	1.25E-183	0
	std	5.67E-17	9.03E-22	0	6.10E-16	0	0
F3	mean	2.83E-05	5.50E-07	0	9.20E-07	0	0
	std	7.43E-05	2.46E-06	0	3.41E-06	0	0
F4	mean	8.68E-07	1.44E-09	6.37E-257	7.15E-07	4.80E-177	4.17E-317
	std	1.07E-06	1.95E-09	0	9.14E-07	0	0
F5	mean	2.71E+01	2.69E+01	2.90E+01	8.46E-01	2.70E+01	1.03E-01
	std	6.86E-01	5.67E-01	1.00E-03	2.27E+00	9.07E-01	2.11E-01
F6	mean	6.54E-01	6.62E-01	6.87E+00	5.77E-05	8.97E-01	3.00E-03
	std	3.42E-01	3.69E-01	3.62E-01	2.31E-05	5.32E-01	6.70E-03
F7	mean	2.10E-03	1.10E-03	8.15E-05	2.40E-03	1.70E-03	5.97E-05
	std	1.20E-03	7.00E-04	8.07E-05	1.30E-03	1.90E-03	7.33E-05
F8	mean	2.67E+00	9.78E-01	0	2.08E-02	0	0
	std	2.60E+00	5.36E+00	0	7.16E-02	0	0
F9	mean	1.02E-13	2.28E-14	8.88E-16	1.40E-13	8.88E-16	8.88E-16
	std	1.92E-14	3.97E-15	0	2.71E-14	0	0
F10	mean	4.10E-03	1.00E-03	0	6.30E-07	0	0
	std	8.30E-03	4.10E-03	0	3.45E-06	0	0

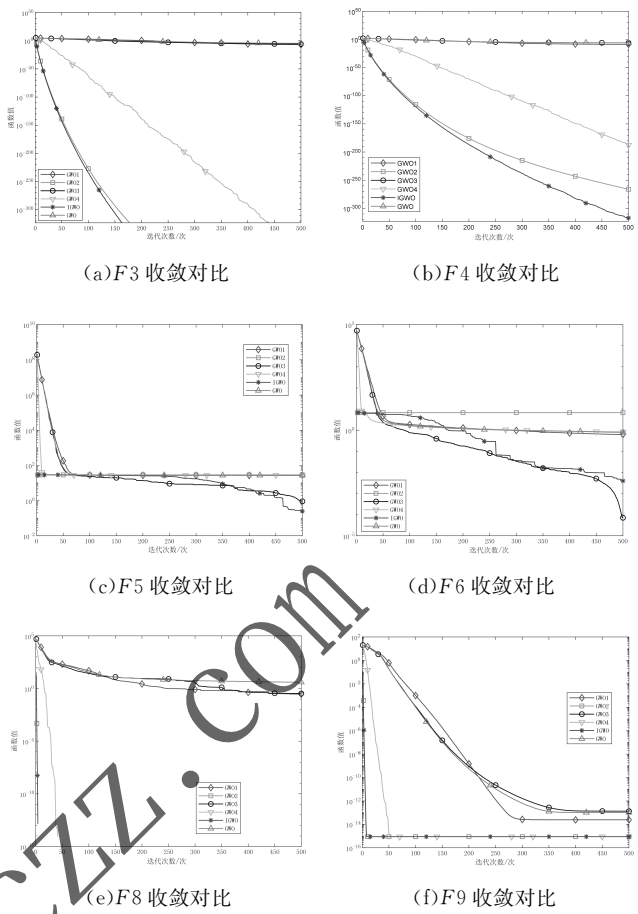


图 4 各改进 GWO 算法的收敛曲线图

Fig. 4 Convergence curves of each improved GWO algorithm

从表 3 可以看出,对于函数 $F1, F3, F8 \sim F10$, GWO2、GWO4 和 IGWO 的平均值比采用其他改进策略的算法更接近理论最优值,甚至在函数 $F1, F3, F8, F10$ 上达到理论最优值。对于函数 $F2, F4, F5, F7$, IGWO 的平均值比其他所有改进算法都要更接近理论最优值。对于函数 $F6$,可以看出 GWO3 算法的寻优能力最强,IGWO 次之。从表 3 中所有基准测试函数的标准差可以看出,IGWO 算法的标准差大多数要低于其他算法,说明改进后的灰狼优化算法进行的 30 次实验的平均数据更集中,改进后算法的结果更准确、更稳定。

从图 4 可以看出,IGWO 算法的性能与其他改进算法相比有明显的提升。与 GWO 相比,GWO1 在收敛速度和求解精度上均有不同程度的提升,表明非线性控制参数策略能提高算法的性能。在函数 $F3 \sim F4, F8 \sim F9$ 上,GWO2 和 GWO4 因其采取的改进策略而表现出较优的全局搜索能力。对于函数 $F5, F6$,GWO3 采取 t -分布概率扰动策略提高了算法跳出局部最优的能力。综合来看,应用各改进策略的算法在性能上均有不同程度的提升,融入各种改进策略的 IGWO 算法具有更强的寻优能力和求解能力。

3.3 与其他参考文献的灰狼优化算法对比

文献[13]提出了一种基于停滞检测的双向搜索灰狼优化算法(DBGWO),文献[14]对杂交策略进行改进并将其融入 GWO 算法中,提出一种改进的杂交策略的自适应灰狼优化算法(AGWO),文献[15]把粒子群算法和灰狼算法相结合,提出

一种混合 GWO 算法(PSO_GWO)。将 IGWO 与上述三种算法在维度为 30 的场景下进行对比,结果如表 4 所示,其中加粗字体为最优测试结果,“—”表示参考文献未做该实验。

表 4 与其他学者改进 GWO 算法的测试结果对比

Tab. 4 Test results comparison with other scholars' improved GWO algorithm

函数	评价	DBGWO	AGWO	PSO_GWO	IGWO
F1	mean	1.92E-270	0	4.83E-119	0
	std	0	0	1.61E-119	0
F2	mean	4.43E-137	3.47E-277	1.34E-59	0
	std	1.85E-137	0	4.18E-59	0
F3	mean	9.52E-239	0	1.22E-113	0
	std	0	0	4.86E-113	0
F4	mean	2.56E-130	3.12E-268	6.03E-61	4.17E-317
	std	4.80E-122	0	2.31E-60	0
F5	mean	7.94E+00	2.74E+01	2.69E+01	1.03E-01
	std	1.16E+01	1.83E-01	1.10E-02	2.11E-01
F6	mean	1.83E-04	5.35E-01	—	3.00E-03
	std	1.01E-04	9.74E-02	—	6.70E-03
F7	mean	1.40E-03	1.23E-04	4.35E-04	5.97E-05
	std	8.22E-04	6.58E-09	3.23E-04	7.33E-05
F8	mean	0	0	0	0
	std	0	0	0	0
F9	mean	2.30E-15	8.88E-16	8.18E-16	8.88E-16
	std	1.77E-15	0	1.72E-15	0
F10	mean	0	0	9.00E-04	0
	std	0	0	3.00E-04	0

从表 4 可以看出,与其他学者提出的改进算法相比,改进后的 IGWO 在 10 个基准测试函数(除 F6、F9 外)上的均值显然更小。同时,对比基准函数的标准差发现,IGWO 的标准差在除了 F5、F6、F7 函数外均位列第一,说明其算法收敛稳定性较强。将 IGWO 分别与其他三种改进算法进行对比,发现 IGWO 在 70% 的测试函数上优于 DBGWO,在 20% 的测试函数上效果相同,仅在 10% 的测试函数上较差;在 30% 的测试函数上优于 AGWO,在 50% 的测试函数上效果相同;在 67% 的测试函数上优于 PSO_GWO。

综上所述,与基本 GWO 算法相比,IGWO 算法具有更好的寻优精度及寻优稳定性。与目前改进的 GWO 算法相比,IGWO 算法的性能更好。

4 结论(Conclusion)

本文提出了一种基于混合扰动的 GWO 算法,该算法在求解复杂优化问题的全局最优解时,具有更好的搜索精度和收敛速度。通过引入控制参数和惯性权重,有利于平衡算法全局搜索和局部搜索的能力,加入 t -分布概率扰动策略和最优灰狼扰动策略,有利于增强算法跳出局部最优的能力。通过对 10 个基准测试函数进行测试,从平均值和标准差可以看出,IGWO 具有更佳的性能。未来的研究工作有两个部分:一是不断改进灰狼优化算法,以期获得性能更好的算法;二是将改进算法应用于实际,如解决具体的工程设计、车间调度等问题。

参考文献(References)

- [1] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [2] KOHLI M, ARORA S. Chaotic grey wolf optimization algorithm for constrained optimization problems[J]. Journal of Computational Design and Engineering, 2018, 5(4): 458-472.
- [3] FU W, WANG K, LI C, et al. Multi-step short-term wind speed forecasting approach based on multi-scale dominant ingredient chaotic analysis, improved hybrid GWO-SCA optimization and ELM[J]. Energy Conversion and Management, 2019, 187(4): 356-377.
- [4] ZHANG X, MIAO Q, LIU Z, et al. An adaptive stochastic resonance method based on grey wolf optimizer algorithm and its application to machinery fault diagnosis[J]. ISA Transactions, 2017, 71: 206-214.
- [5] 王正通,程凤芹,尤文,等.基于翻筋斗觅食策略的灰狼优化算法[J]. 计算机应用研究, 2021, 38(5): 1434-1437.
- [6] 徐明,龙文.基于多策略融合灰狼优化算法的特征选择方法[J]. 科学技术与工程, 2021, 21(20): 8544-8551.
- [7] MA C, HUANG H, FAN Q, et al. Grey wolf optimizer based on Aquila exploration method[J]. Expert Systems with Applications, 2022, 205: 117629.
- [8] 邢燕祯,王东辉.一种基于收敛因子改进的灰狼优化算法[J]. 网络新媒体技术, 2020, 9(3): 28-34.
- [9] 魏政磊,赵辉,李牧东,等.控制参数值非线性调整策略的灰狼优化算法[J]. 空军工程大学学报(自然科学版), 2016, 17(3): 68-72.
- [10] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization[J]. Swarm Intelligence, 2007, 1(1): 33-57.
- [11] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [12] DHIMAN G, KUMAR V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems [J]. Knowledge-based Systems, 2019, 165: 169-196.
- [13] 张大明,徐嘉庆,赵彦清,等.基于停滞检测的双向搜索灰狼优化算法[J]. 计算机应用研究, 2022, 39(6): 1725-1730.
- [14] 刘紫燕,吴应雨,梁静,等.基于杂交策略的自适应灰狼优化算法[J]. 计算机应用研究, 2022, 39(1): 113-117.
- [15] TENG Z, LV J, GUO L. An improved hybrid grey wolf optimization algorithm[J]. Soft Computing, 2019, 23(15): 6617-6631.

作者简介:

李征南(1998-),男,硕士生。研究领域:智能优化算法。

秦江涛(1966-),男,博士,副教授。研究领域:制造系统分析。