

一种新型非线性收敛因子的灰狼优化算法*

王敏^{1,2}, 唐明珠³

(1. 湖南机电职业技术学院 信息工程学院, 长沙 410151; 2. 湖南大学 计算机与通信学院, 长沙 410082; 3. 长沙理工大学 能源与动力工程学院, 长沙 410114)

摘要: 针对标准灰狼优化算法在求解复杂工程优化问题时存在求解精度不高和易陷入局部最优的缺点, 提出一种新型灰狼优化算法用于求解无约束连续函数优化问题。该算法首先利用反向学习策略产生初始种群个体, 为算法全局搜索奠定基础; 受粒子群优化算法的启发, 提出一种非线性递减收敛因子更新公式, 其动态调整以平衡算法的全局搜索能力和局部搜索能力; 为避免算法陷入局部最优, 对当前最优灰狼个体进行变异操作。对 10 个测试函数进行仿真实验, 结果表明, 与标准灰狼优化算法相比, 改进灰狼优化算法具有更好的求解精度和更快的收敛速度。

关键词: 灰狼优化算法; 反向学习策略; 函数优化; 非线性

中图分类号: TP301.6

文献标志码: A

文章编号: 1001-3695(2016)12-3648-06

doi:10.3969/j.issn.1001-3695.2016.12.029

Novel grey wolf optimization algorithm based on nonlinear convergence factor

Wang Min^{1,2}, Tang Mingzhu³

(1. Dept. of Information Engineering, Hunan Mechanical & Electrical Polytechnic, Changsha 410151, China; 2. School of Computer & Communication, Hunan University, Changsha 410082, China; 3. School of Energy & Power Engineering, Changsha University of Science Engineering, Changsha 410114, China)

Abstract: The classical grey wolf optimization (GWO) algorithm has a few disadvantages of low solving precision and high possibility of being trapped in local optimum. This paper proposed a novel grey wolf optimization (NGWO) algorithm for solving unconstrained optimization problems. The proposed algorithm used opposition-based learning strategy to initiate population, which strengthened the diversity of global searching. Inspired by particle swarm optimization (PSO), this paper proposed an improved convergence factor update equation, which was based on that the values of parameter a are nonlinearly decreased over the course of iterations. The convergence factor was dynamically adjusted to maintain a better balance between global search and local search. Mutation operator was given on the current optimal individual of each generation, thus it could effectively jump out of local minima. Experiments are conducted on a set of 10 unconstrained benchmark functions. Based on the results, the proposed NGWO algorithm shows significantly better performance than the standard GWO algorithm.

Key words: grey wolf optimization algorithm; opposition-based learning strategy; function optimization; nonlinear

0 引言

在工程设计、结构优化等实际工程应用和科学研究中存在大量的优化问题, 这些问题可转换为求解数值的无约束优化问题。不失一般性, 一个非线性无约束优化问题(最小化)可描述为^[1]

$$\begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & l_i \leq x_i \leq u_i, i=1, 2, \dots, d \end{aligned} \quad (1)$$

其中: $f(x)$ 为目标函数, $x = (x_1, x_2, \dots, x_d)$ 为 d 维决策变量, l_i 和 u_i 分别为变量 x_i 的上限和下限。

由于问题(1)通常具有强非线性、复杂、高维的搜索空间, 基于梯度信息的传统优化方法难以对其进行有效的求解^[2]。在过去的几十年中, 受自然界一些现象启发, 智能随机搜索算法如遗传算法(genetic algorithm, GA)、差分进化(differential evolution, DE)算法、人工蜂群算法(artificial bee colony, ABC)、蚁群算

法(artificial colony optimization, ACO)、粒子群优化(particle swarm optimization, PSO)算法、萤火虫(glowworm swarm optimization, GSO)算法、引力搜索算法(gravitational search algorithm, GSA)等具有原理简单、容易实现、能以较大概率收敛到问题的全局最优解, 因此在无约束优化问题中有着广泛的应用^[3-8]。

灰狼优化(grey wolf optimization, GWO)算法^[9]是 Mirjalili 于 2014 年提出的一种新型群体智能优化方法, 它源于模拟自然界中灰狼种群的等级层次机制和捕食行为, 通过狼群跟踪、包围、追捕、攻击猎物等过程实现优化搜索目的。GWO 算法具有原理简单、易于实现, 需调整的参数少且不需要问题的梯度信息, 在函数优化方面, GWO 算法已被证明在求解精度和稳定性上要明显优于 PSO 算法、DE 算法和 GSA 算法^[9]。因此, GWO 算法在经济调度指派问题^[10]、K-均值聚类优化^[11]、多输入多输出电力系统^[12]、面波参数估计^[13]、直流电机最优控制^[14]、流水线车间调度问题^[15]、多层传感器训练^[16]等领域中有

收稿日期: 2016-01-06; **修回日期:** 2016-03-07 **基金项目:** 国家自然科学基金资助项目(61403046); 湖南省科学计划资助项目(2014FJ3051)

作者简介: 王敏(1978-), 男, 湖南浏阳人, 副教授, 博士研究生, 主要研究方向为进化计算、智能优化算法(wangmin_7819@126.com); 唐明珠(1983-), 男, 讲师, 博士, 主要研究方向为进化计算、故障诊断。

着广泛的应用。

尽管 GWO 算法在许多领域中获得了较成功的应用,但与其他基于种群迭代的智能搜索算法相似,GWO 算法也存在解精度低、难以协调其勘探和开采能力的缺点。为了减少 GWO 算法陷入局部最优的概率,Zhu 等人^[17]利用 DE 算法强大的全局搜索能力,提出一种混合 GWO 算法用于函数优化;为了提高解精度和局部搜索能力,龙文等人^[18]采用佳点集方法产生初始个体,对当前最优灰狼个体执行 Powell 局部搜索操作,提出一种改进 GWO 用于约束函数优化;为增强 GWO 算法的全局搜索能力,Saremi 等人^[19]结合动态进化种群技术,提出一种混合 GWO 算法用于求解函数优化问题。为改善标准 GWO 算法的寻优性能,本文引入反向学习策略产生初始个体以维持灰狼初始种群的多样性;受 PSO 算法中惯性权重的启发,提出一种随迭代次数非线性递减的自适应收敛因子以平衡算法的全局搜索和局部搜索能力;引入变异算子以减少算法发生早熟收敛的概率。

1 基本灰狼优化算法

灰狼优化算法是通过模拟自然界中灰狼群体的社会等级机制和捕食行为而提出的一种新型群体搜索方法。灰狼群体的社会等级机制如图 1 所示。

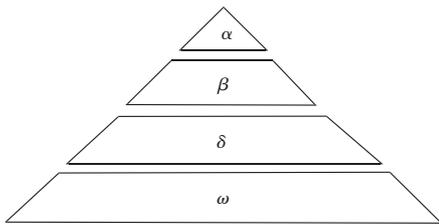


图 1 灰狼群体的社会等级机制

在图 1 中,灰狼群体的社会等级层次分为四层:α、β、δ 和 ω 层,其中 α 称为头狼,领导灰狼群体;β 是下属的灰狼,协助头狼 α 作出决策;δ 称为普通狼,听从 α 及 β 的指令,也可以指挥其他底层狼 ω。灰狼群体在捕获猎物时,其他灰狼个体在头狼 α 的带领下有组织地对猎物进行围攻。首先,狼群通过气味等信息追踪猎物并逐渐靠近;然后,在确定猎物位置后,狼群包围猎物;最后,逐渐缩小包围圈,攻击猎物。

在 D 维搜索空间中,假设 N 只灰狼个体组成种群 $X = (X_1, X_2, \dots, X_N)$,定义第 i 只灰狼的位置为 $X_i = (X_i^1, X_i^2, \dots, X_i^D)$,其中 X_i^d 表示第 i 只灰狼在第 d 维上的位置。定义群体历史最优解为头狼 α,历史次最优解为下属狼 β,历史第三最优解为普通狼 δ,种群中其他个体为 ω。首先描述灰狼逐渐接近并包围猎物的行为,对第 i 只灰狼:

$$X_i^d(t+1) = X_p^d(t) - A_i^d |C_i^d X_p^d(t) - X_i^d(t)| \quad (1)$$

其中: t 为当前迭代次数, $X_p = (X_p^1, X_p^2, \dots, X_p^D)$ 为猎物位置, $A_i^d |C_i^d X_p^d(t) - X_i^d(t)|$ 为包围步长, A_i^d 和 C_i^d 分别为

$$A_i^d = 2a \cdot \text{rand}_1 - a \quad (2)$$

$$C_i^d = 2 \cdot \text{rand}_2 \quad (3)$$

$\text{rand}_1, \text{rand}_2$ 表示 $[0, 1]$ 间的随机变量; a 称为收敛因子,随迭代次数的增大从 2 线性减小到 0:

$$a = 2 - t/t_{\max} \quad (4)$$

其中: t_{\max} 为最大迭代次数。

灰狼群体根据 α、β 和 δ 的位置 X_α, X_β 和 X_δ 进行更新各自

的位置:

$$\begin{cases} X_{i,\alpha}^d(t+1) = X_\alpha^d(t) - A_{i,1}^d |C_{i,1}^d X_\alpha^d(t) - X_i^d(t)| \\ X_{i,\beta}^d(t+1) = X_\beta^d(t) - A_{i,2}^d |C_{i,2}^d X_\beta^d(t) - X_i^d(t)| \\ X_{i,\delta}^d(t+1) = X_\delta^d(t) - A_{i,3}^d |C_{i,3}^d X_\delta^d(t) - X_i^d(t)| \end{cases} \quad (5)$$

$$X_i^d(t+1) = \sum_{j=\alpha,\beta,\delta} w_j X_{i,j}^d(t+1) \quad (6)$$

其中: $w_j(j = \alpha, \beta, \delta)$ 表示 α、β 和 δ 的权重系数:

$$w_j = \frac{f(X_j(t))}{f(X_\alpha(t)) + f(X_\beta(t)) + f(X_\delta(t))} \quad (7)$$

式中: $f(X_j(t))$ 表示第 j 只灰狼个体在第 t 代的适应度值。

2 新型灰狼优化算法

2.1 反向学习初始化种群

Haupt 等人^[20]指出,对基于种群迭代的群体智能优化算法来说,初始种群的好坏影响着算法的全局搜索速度和解的质量,多样性较好的初始种群对提高算法的寻优性能很有帮助。然而,标准 GWO 算法在迭代前采用随机初始化种群个体,难以确保初始种群的多样性,从而在一定程度上影响了算法的搜索效率。反向学习策略^[21]是由 Tizhoosh 于 2005 年提出的,目前已在 GA、DE、ACO 和 BBO 等群体智能优化算法中得到了成功的应用。因此,本文采用反向学习策略来产生初始种群。

定义 1 反向点。假设在 $[l, u]$ 上存在数 x ,则 x 的反向点定义为 $x' = l + u - x$ 。将反向点的定义扩展到 D 维空间,设 $p = (x_1, x_2, \dots, x_D)$ 为 D 维空间中的一个点,其中 $x_i \in [l_i, u_i], i = 1, 2, \dots, D$,则其反向点 $p' = (x'_1, x'_2, \dots, x'_D)$,其中 $x'_i = l_i + u_i - x_i$ 。

根据上述定义,采用反向学习策略来产生初始种群个体的步骤如下:

- 在搜索空间中随机初始化 N 个灰狼个体位置 $x_{i,j}(i = 1, 2, \dots, D; j = 1, 2, \dots, N)$ 作为初始种群 RP;
- 根据定义 1,初始种群 RP 中的每个灰狼个体 x_i 的反向个体 x'_i 构成反向种群 OP;
- 合并种群 RP 和 OP,将其 $2N$ 个灰狼个体按照适应度值进行升序排序,选取适应度值前 N 个灰狼个体作为初始种群。

2.2 收敛因子非线性调整策略

任何一种群体智能优化算法在进化过程中均存在全局搜索和局部搜索两类不同的操作,若这两类操作协调不好,则使算法陷入局部最优或降低其收敛性能的概率增大。强的全局搜索能力能维持种群多样性,避免算法陷入局部最优;强的局部搜索能力能保证算法进行局部精确搜索,加快算法的收敛速度。因此,作为群体智能算法之一,GWO 算法能否获得高寻优性能的关键是如何在全局搜索和局部搜索能力之间进行有力的协调。由文献[9]可知,当 $|A| > 1$ 时,灰狼群体将扩大包围圈,以寻找更好的猎物,此时对应于全局搜索;当 $|A| < 1$ 时,灰狼群体将收缩包围圈,以对猎物完成最后的攻击行为,此时对应于局部精确搜索。因此, A 值的大小与 GWO 算法的全局搜索和局部搜索能力有很大关系。另外,由式(2)和(4)可知, A 随着收敛因子 a 的变化而不断变化,且收敛因子 a 随着迭代次数的增加从 2 线性递减到 0。

然而,GWO 算法在搜索过程中却是非线性变化的,收敛因子 a 线性递减策略不能完全体现出实际的优化搜索过程。因此,本文提出一种非线性变化收敛因子更新公式:

$$a(t) = a_{\text{initial}} + (a_{\text{final}} - a_{\text{initial}}) \times \left(1 - \frac{t}{t_{\text{max}}}\right)^{k_2} \quad (8)$$

其中: a_{initial} 和 a_{final} 分别为收敛因子 a 初始值和终止值, t 为当前迭代次数, t_{max} 为最大迭代次数, k_1 和 k_2 为非线性调节系数。由式(8)可知, 收敛因子 a 随进化迭代次数增加而非线性动态变化, 从而为有效地平衡算法的全局搜索能力和局部搜索能力提供保证。

2.3 变异算子

基于最优保存策略, 在 GWO 算法的进化后期, 由于群体中所有灰狼个体均向最优个体区域逼近, 从而导致群体多样性损失, 此时, 如果当前最优灰狼个体为局部最优解, 则 GWO 算法陷入局部最优, 出现早熟收敛现象, 这也是群体智能优化算法的固有特点。为了降低 GWO 算法出现早熟现象和陷入局部最优的概率, 本文对群体中当前最优灰狼个体进行多样性变异操作, 其原理如下^[22]:

假设某个体 $x_i = (x_1, x_2, \dots, x_D)$, 以概率 $1/D$ 随机从个体 x_i 中选取一个元素 $x_k (k=1, 2, \dots, D)$, 然后在 $[l_i, u_i]$ 范围内随机产生一个实数替代个体 x_i 中的元素 x_k , 从而产生一个新的个体 $x'_i = (x'_1, x'_2, \dots, x'_D)$ 。多样性变异算子为

$$x'_i = \begin{cases} l_i + \lambda \cdot (u_i - l_i) & i = k \\ x_i & \text{otherwise} \end{cases} \quad (9)$$

其中: l_i 和 u_i 分别为变量 x_i 的上界和下界, $\lambda \in [0, 1]$ 为随机数。

2.4 NGWO 算法步骤

- a) 设置算法参数, 种群规模 N , 最大迭代次数 t_{max} , 收敛因子初始值 a_{initial} 和终止值 a_{final} , 调节系 k_1, k_2 。
- b) 在搜索空间中利用 2.1 节所描述的反向学习策略产生 N 个个体作为初始种群, 令 $t = 1$ 。
- c) 计算群体中每个个体的适应度值并排序, 将适应度值排列前三位的个体位置分别记为 X_α, X_β 和 X_δ 。
- d) 判断算法是否满足终止条件, 若满足, 输出最优灰狼个体, 则算法结束; 否则, 执行步骤 e)。
- e) 根据式(8)计算出收敛因子 a 的值, 然后根据式(2)和(3)计算出 A 和 C 。
- f) 根据式(5)和(6)更新群体中每个灰狼个体的位置。
- g) 对当前群体中最优灰狼个体执行 2.3 节所描述的变异操作以产生新的灰狼个体, 令 $t = t + 1$, 返回步骤 c)。

3 仿真实验及分析

3.1 测试函数

为了检验本文提出的改进灰狼优化(记为 NGWO)算法的寻优性能, 从文献[9]中选取 10 个标准测试函数进行仿真实验, 10 个测试函数的具体特征如表 1 所示, 其中, $f_1 \sim f_7$ 为复杂单峰优化问题, $f_8 \sim f_{10}$ 为多峰优化问题。

表 1 10 个测试函数

函数名	维数	函数表达式	搜索区间	f_{min}	收敛精度
Sphere	30	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0	1×10^{-8}
Schwefel 2.22	30	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0	1×10^{-8}
Schwefel 1.2	30	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]$	0	1×10^{-8}
Schwefel 2.21	30	$f_4(x) = \max_i \{ x_i , 1 \leq x_i \leq n \}$	$[-100, 100]$	0	1×10^{-8}
Rosenbrock	30	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0	1×10^0
Step	30	$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]$	0	1×10^{-1}
Quartic	30	$f_7(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1]$	$[-1.28, 1.28]$	0	1×10^{-4}
Rastrigin	30	$f_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0	1×10^{-8}
Ackley	30	$f_9(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	0	1×10^{-8}
Griewank	30	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0	1×10^{-8}

3.2 与 GWO 和 HGWO 算法的比较

利用 NGWO 算法对 10 个标准测试函数进行求解, 并与标准 GWO 算法、Hybrid differential evolution and grey wolf optimization (HGWO) 算法^[17] 进行比较。NGWO 算法与标准 GWO 算法进行比较是说明本文所提出的改进策略的有效性。HGWO 算法是一种混合算法, 即混合差分进化(DE)和 GWO 算法, 其寻优性能比标准 GWO 算法有了较大的提高。NGWO 算法与 HGWO 算法进行比较是为了进一步说明本文算法的竞争力, 避免只与标准 GWO 算法比较结果的可信度。为了比较结果的公平性, 三种算法的参数设置如下: 种群规模均设置为 30, 最大迭代次数均设为 500。10 个测试函数在上述参数设置的条件下, 采用 GWO 算法和 NGWO 算法分别独立运行 30 次实验, 记录三种算法的最优值、平均值、最差值和标准差, 比较结果如表 2 所示, 其中, HGWO 算法的结果直接来源于参考文献[10]。

从表 2 中的比较结果可知, 在满足固定收敛精度条件下, 除了 Rosenbrock 函数(f_5), 本文提出的 NGWO 算法在其他九个测试函数上进行 30 次实验中均能一致收敛到问题的全局最优解, 尤其是 Rastrigin 函数(f_8) 和 Griewank 函数(f_{10}), NGWO

算法均能收敛到理论最优值 0。对于 Quartic(f_7) 函数, NGWO 算法的寻优成功率为 60%, 其余八个测试函数的寻优成功率均为 100% (求解结果满足固定收敛精度就说明该算法收敛)。标准 GWO 算法对于 Sphere 函数(f_1)、Schwefel2.22 函数(f_2) 和 Ackley 函数(f_{10}) 30 次实验均能一致收敛全局最优解, 寻优成功率为 100%; 对于其他七个测试函数 Schwefel1.2、Schwefel2.21、Rosenbrock、Step、Quartic、Rastrigin 和 Griewank, 标准 GWO 算法的寻优成功率分别为 10%、0%、0%、80%、20%、50% 和 70%。

与标准 GWO 算法相比, NGWO 算法在八个函数($f_1, f_2, f_3, f_4, f_6, f_7, f_8, f_9$) 上获得了较好的寻优结果; 对于函数 f_5 , GWO 和 NGWO 算法获得了相似的结果; 对于函数 f_{10} , 两种算法获得了相似的最优结果, 另外, NGWO 算法取得了较好的平均值、最差值和标准差。

与 HGWO 算法相比, NGWO 算法在六个函数($f_1, f_2, f_3, f_4, f_7, f_9$) 上获得了较好的寻优结果; 对于函数 f_5 , HGWO 算法获得了较好的最优值和平均值, 而 NGWO 算法取得了较好的最差值和标准差; 对于 f_6 函数, HGWO 算法得到了较好的寻优结果; 对于函数 f_8 和 f_{10} , 两种算法获得了相似的最优值, 而 NG-

WO 取得了较好的平均值、最差值和标准差值。由上述比较结果可以看出,NGWO 算法比 GWO 算法和 HGWO 算法在大部分测试函数上具有更高的求解精度。

表 2 三种算法对 10 个测试函数的结果比较

函数	算法	最优值	平均值	最差值	标准差
f_1	GWO	2.52E-29	1.07E-27	5.03E-27	1.88E-27
	HGWO	2.92E-34	1.12E-32	8.95E-32	2.32E-32
	NGWO	5.92E-49	1.16E-47	8.72E-47	2.72E-47
f_2	GWO	3.67E-17	7.94E-17	2.25E-16	5.55E-17
	HGWO	1.65E-20	9.33E-20	3.60E-19	6.92E-20
	NGWO	4.03E-29	2.92E-28	1.33E-27	4.16E-28
f_3	GWO	3.91E-08	2.07E-05	5.46E-04	1.85E-05
	HGWO	6.07E-11	3.18E-08	3.08E-07	6.55E-08
	NGWO	2.03E-10	9.98E-12	6.74E-11	1.99E-11
f_4	GWO	1.02E-07	6.46E-07	2.11E-06	5.67E-07
	HGWO	5.81E-09	4.17E-08	2.39E-07	4.56E-08
	NGWO	2.97E-14	7.15E-13	2.34E-12	7.60E-13
f_5	GWO	26.0066	27.0096	27.9430	0.54298
	HGWO	25.2230	26.4876	28.5405	0.70271
	NGWO	25.9421	26.0516	27.1280	0.39602
f_6	GWO	2.53E-01	6.64E-01	1.23E+00	2.92E-01
	HGWO	2.26E-05	3.78E-01	7.53E-01	2.26E-01
	NGWO	1.94E-01	5.62E-01	8.88E-01	2.32E-01
f_7	GWO	4.13E-04	1.95E-03	4.22E-03	1.21E-03
	HGWO	3.66E-04	1.49E-03	3.27E-03	7.53E-04
	NGWO	2.34E-04	1.05E-03	1.51E-03	2.32E-04
f_8	GWO	5.68E-14	3.20E+00	1.23E+01	4.00E+00
	HGWO	0	2.27E-01	4.7666	9.20E-01
	NGWO	0	0	0	0
f_9	GWO	7.55E-14	1.00E-13	1.75E-13	5.90E-14
	HGWO	3.64E-14	4.27E-14	5.06E-14	4.37E-15
	NGWO	7.99E-15	1.05E-14	1.51E-14	2.39E-15
f_{10}	GWO	0	7.19E-03	4.31E-02	1.40E-02
	HGWO	0	1.37E-03	3.12E-02	5.82E-03
	NGWO	0	0	0	0

对五个函数的寻优收敛曲线。从图 2 可以清晰地看出,对于五个函数,NGWO 算法均能比 GWO 算法具有较快的收敛速度和较高的收敛精度。

由标准 GWO 算法的机理可知,算法的勘探和开发能力取决于收敛因子 a ,而收敛因子 a 随进化迭代次数线性变化。另外,由第 2 章可以看出,本文主要改进策略是收敛因子 a 随进化迭代次数非线性动态变化。为了进一步说明非线性动态变化收敛因子策略的有效性,选取 10 个测试函数进行数值实验。在这里,将含有线性变化收敛因子的灰狼优化算法记为 GWO,将含有非线性动态变化收敛因子的灰狼优化算法记为 GWO-1。为了进行公平的比较,两种算法采用了相同的参数设置。表 3 给出了两种算法 30 次实验的平均值和标准差结果比较。

由表 3 可以看出,与 GWO 算法相比,GWO-1 算法在 10 个测试函数上均获得了较好的寻优结果,这充分说明了非线性动态变化的收敛因子可以改进 GWO 算法的性能。

表 3 两种算法对 10 个测试函数的结果比较

函数	GWO 算法		GWO-1 算法	
	平均值	标准差	平均值	标准差
f_1	1.07E-27	1.88E-27	1.47E-46	1.65E-46
f_2	7.94E-17	5.55E-17	7.56E-27	6.30E-27
f_3	2.07E-05	1.85E-05	1.60E-10	1.98E-10
f_4	6.46E-07	5.67E-07	2.78E-12	2.06E-12
f_5	27.0096	0.54298	27.0044	0.42844
f_6	6.64E-01	2.92E-01	6.51E-01	1.62E-01
f_7	1.95E-03	1.21E-03	1.18E-03	5.34E-04
f_8	3.20E+00	4.00E+00	0	0
f_9	1.00E-13	5.90E-14	1.30E-14	3.18E-15
f_{10}	7.19E-03	1.40E-02	0	0

3.3 与其他智能搜索算法的比较

为了进一步验证 NGWO 算法的有效性,将其与文献[23]中九种有代表性的智能搜索算法,即 Comprehensive learning PSO (CLPSO)算法、Orthogonal learning PSO (OLPSO)算法、Hierarchical PSO (HPSO)算法、Self-adaptive DE (SaDE)算法、Adaptive DE (JADE)算法、Self-adaptive control parameter in DE (jDE)算法、Gbest-guided ABC (GABC)算法、Novel search equation ABC (CABC)算法和 Orthogonal learning ABC (OCABC)算法的结果进行比较,比较结果如表 4~6 所示。表中九种算法的参数设置和寻优结果直接来源于参考文献,表中黑粗体为比较算法中的最好结果。为了比较结果的公平性,比较算法均采用相同的适应度函数计算次数,九种算法的最大适应度函数计算次数如表 4~6 所示。

由表 4 中比较结果可知,在最大适应度函数计算次数为 200 000 次的条件下,NGWO 算法在三个函数(f_1 、 f_8 和 f_{10})上均获得了理论最优值 0;对于函数 f_2 ,NGWO 算法虽然没有取得理论最优值,但其 30 次实验的均值 2.46e-260 非常接近全局最优解 0。与 HPSO 算法相比,除了函数 f_6 ,GWO 算法在其他七个函数上均获得了较好的平均值和标准差。与 CLPSO 算法相比,NGWO 算法在函数 f_1 、 f_2 、 f_7 、 f_8 、 f_9 和 f_{10} 上获得了较好的平均值和标准差;然而,对于函数 f_5 和 f_6 ,CLPSO 算法得到了较好的寻优结果。与 OLPSO 算法相比,NGWO 算法在五个函数(f_1 、 f_2 、 f_7 、 f_8 和 f_{10})上取得了较好的寻优结果和两个函数(f_5 和 f_9)上获得了相似的结果;对于函数 f_6 ,OLPSO 算法获得了较好的寻优结果。

由表 5 可知,在最大适应度函数计算次数为 150 000 次的条件下,NGWO 算法在三个测试函数(f_1 、 f_8 和 f_{10})上取得了理

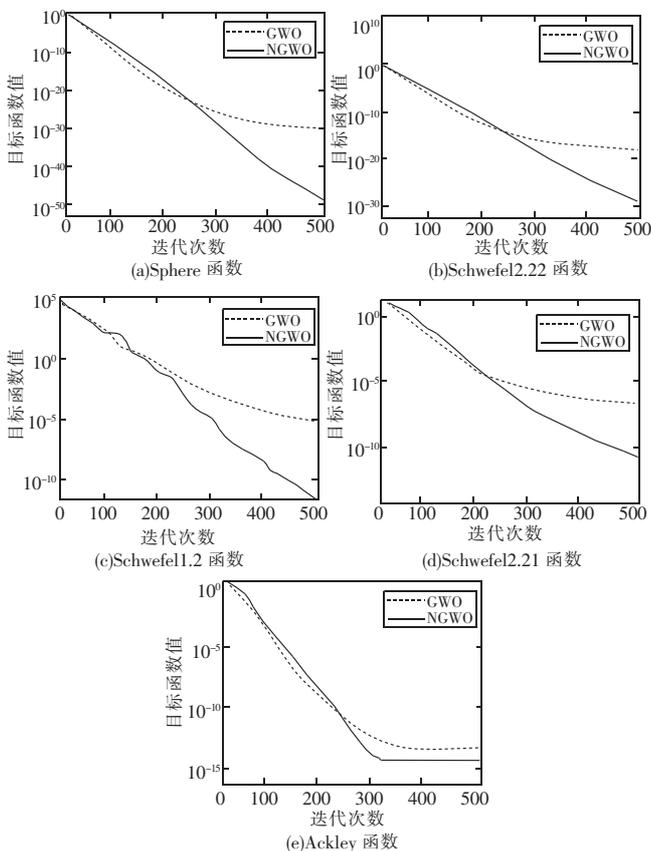


图 2 GWO 和 NGWO 算法对五个函数的收敛曲线

图 2 给出了 GWO 算法和 NGWO 算法在固定迭代次数下

论最优值 0;对于函数 f_2 和 f_4 ,NGWO 算法在 30 次实验中得到的平均值非常接近全局最优解。与 GABC 算法相比,NGWO 算法在七个测试函数($f_1, f_2, f_4, f_7, f_8, f_9$ 和 f_{10})上获得了较好的寻优结果;然而,GABC 算法则在函数 f_5 和 f_6 上得到了较好的平均值和标准差。与 CABC 相比,NGWO 在函数 $f_1, f_2, f_4, f_7, f_9, f_{10}$ 上获得了较好的结果和函数 f_8 上取得了相似的寻优结果;对于函数 f_5 和 f_6 ,CABC 算法得到了较好的寻优结果。与

OCABC 算法相比,NGWO 算法在四个函数(f_1, f_2, f_4, f_7)和三个测试函数(f_8, f_9, f_{10})上分别获得了较好和相似的平均值和标准差;对于函数 f_5 和 f_6 ,OCABC 取得了较好的结果。

由表 6 中结果可知,在相同的适应度函数计算次数的条件下,NGWO 算法在全部八个测试函数上获得的结果要比 SaDE 算法均要优;与 JADE 和 jDE 算法相比,除了函数 f_5 ,NGWO 算法在其他七个测试函数上均取得了较好的平均值和标准差。

表 4 NGWO 算法与 HPSO、CLPSO、OLPSO 算法的结果比较

函数	适应度计算次数 (FEs)	HPSO		CLPSO		OLPSO		NGWO	
		平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	200 000	2.83E-33	3.19E-33	1.58E-12	7.70E-13	4.12E-54	6.34E-54	0	0
f_2	200 000	9.03E-20	9.58E-20	2.51E-08	5.84E-09	3.51E-25	2.74E-25	2.46E-260	0
f_5	200 000	2.39E+01	2.65E+01	1.13E+01	9.85E+00	2.15E+01	2.99E+01	2.61E+01	7.00E-01
f_6	200 000	0	0	0	0	0	0	2.39E-01	1.88E-01
f_7	200 000	9.82E-02	3.26E-02	5.85E-03	1.11E-03	1.16E-02	4.10E-03	8.94E-05	6.04E-05
f_8	200 000	9.43E+00	3.48E+00	9.09E-05	1.25E-04	1.07E+00	9.92E-01	0	0
f_9	200 000	7.29E-14	3.00E-14	3.66E-07	7.57E-08	7.98E-15	2.03E-15	7.99E-15	0
f_{10}	200 000	9.75E-03	8.33E-03	9.02E-09	8.57E-09	4.83E-03	8.63E-03	0	0

表 5 NGWO 算法与 GABC、CABC、OCABC 算法的结果比较

函数	适应度计算次数 (FEs)	GABC		CABC		OCABC		NGWO	
		平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	150 000	1.92E-22	1.16E-22	5.41E-35	7.02E-35	4.32E-43	8.16E-43	0	0
f_2	150 000	3.23E-12	9.27E-13	1.43E-18	5.68E-19	1.17E-22	7.13E-23	2.42E-192	0
f_4	150 000	7.00E-00	1.01E-00	6.01E-00	8.80E-01	5.67E-01	2.73E-01	3.14E-87	4.34E-87
f_5	150 000	1.15E-00	2.81E-00	1.96E-01	1.36E-01	7.89E-01	6.27E-01	2.61E+01	5.19E-01
f_6	150 000	0	0	0	0	0	0	4.45E-01	2.55E-01
f_7	150 000	8.94E-02	2.42E-02	4.98E-02	1.78E-02	4.39E-03	2.03E-03	9.64E-05	5.64E-05
f_8	150 000	1.15E-15	3.22E-15	0	0	0	0	0	0
f_9	150 000	2.15E-11	1.06E-11	3.07E-14	2.88E-15	5.32E-15	1.82E-15	7.99E-15	0
f_{10}	150 000	2.08E-03	6.54E-03	4.93E-04	2.20E-03	0	0	0	0

表 6 NGWO 算法与 SaDE、JADE、jDE 算法的结果比较

函数	适应度计算次数 (FEs)	SaDE		JADE		jDE		NGWO	
		平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	150 000	3.28E-20	3.63E-20	2.69E-56	1.41E-55	1.46E-28	1.78E-28	0	0
f_2	200 000	3.51E-25	2.74E-25	3.18E-25	2.05E-24	9.02E-24	6.01E-24	2.46E-260	0
f_5	300 000	2.10E+01	7.80E+00	3.20E-01	1.10E+00	1.30E+01	1.40E+01	2.57E+01	4.78E-01
f_6	10 000	5.07E+01	1.34E+01	5.62E+00	1.87E+00	6.13E+02	1.72E+02	1.01E+00	3.88E-01
f_7	300 000	4.86E-03	5.21E-04	6.14E-04	2.55E-04	3.35E-03	8.68E-04	4.43E-05	3.07E-05
f_8	100 000	2.43E+00	1.60E+00	1.33E-01	9.74E-02	3.32E-04	6.39E-04	0	0
f_9	50 000	3.81E-06	8.26E-07	3.35E-09	2.84E-09	2.37E-04	7.10E-05	7.99E-15	0
f_{10}	50 000	2.52E-09	1.24E-08	1.57E-08	1.09E-07	7.29E-06	1.05E-05	0	0

3.4 参数对算法性能影响的分析

在 GWO 算法中,收敛因子 a 的作用是平衡算法的全局搜索能力和局部搜索能力。由式(8)可知,NGWO 算法通过随进化迭代次数增加而非线性变化的收敛因子 a 协调其全局和局部搜索能力。然而,在式(8)中,非线性调节系数 k_1 和 k_2 是关键参数,主要用来控制收敛因子变化的大小,对算法的性能有一定的影响。

本小节通过对非线性调节系数 k_1 和 k_2 选取五组不同的值,即 $k_1=1, k_2=2, k_1=1, k_2=3, k_1=2, k_2=1, k_1=2, k_2=2$ 和

$k_1=2, k_2=3$ 进行数值实验来分析对 NGWO 算法性能的影响。表 7 给出了五组不同的非线性调节系数 k_1 和 k_2 值对 NGWO 算法影响结果比较,其中,黑粗体表示比较算法中的最好结果。

由表 7 中比较结果可知,当非线性调节系数 $k_1=2$ 和 $k_2=1$ 时,NGWO 算法的寻优性能最佳。对于函数 f_5, f_7 和 f_9, k_1 和 k_2 取五组不同的值时对 NGWO 算法的数值结果影响不大。对于函数 $f_{10}, k_1=2, k_2=1$ 与 $k_1=2, k_2=2$ 时 NGWO 算法均能收敛到理论最优值 0,要优于其他三组 k_1 和 k_2 值所获得的结果。

表 7 NGWO 算法中不同的 k_1 和 k_2 值的寻优性能比较

函数	$k_1=1, k_2=2$		$k_1=1, k_2=3$		$k_1=2, k_2=1$		$k_1=2, k_2=2$		$k_1=2, k_2=3$	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	3.15E-39	6.32E-39	1.29E-35	2.07E-35	1.16E-47	2.72E-47	4.86E-44	7.80E-44	2.51E-41	2.10E-41
f_2	4.27E-24	2.28E-24	7.44E-22	5.31E-22	3.65E-28	4.16E-28	3.89E-26	2.86E-26	6.07E-25	4.73E-25
f_3	7.74E-08	1.66E-07	2.09E-07	5.34E-07	9.98E-12	1.99E-11	6.65E-09	1.86E-08	1.71E-08	4.39E-08
f_4	8.78E-10	9.18E-10	1.08E-08	8.73E-09	7.15E-13	7.60E-13	3.27E-11	3.03E-11	1.55E-10	2.87E-10
f_5	2.70E+01	8.49E-01	2.70E+01	3.73E-01	2.61E+01	3.96E-01	2.66E+01	4.01E-01	2.67E+01	4.90E-01
f_6	1.04E+00	3.56E-01	1.24E+00	2.73E-01	5.62E+01	2.32E-01	9.50E-01	3.80E-01	7.36E-01	3.34E-01
f_7	1.11E-03	4.84E-04	2.17E-03	7.96E-04	1.05E-03	2.21E-04	1.13E-03	8.77E-04	1.42E-03	1.01E-03
f_8	3.40E+00	5.03E+00	4.06E+00	5.95E+00	0	0	5.68E-15	1.80E-14	4.96E-01	1.57E+00
f_9	3.25E-14	4.27E-15	4.38E-14	2.50E-14	1.05E-14	2.39E-15	1.69E-14	4.51E-15	2.61E-14	3.53E-15
f_{10}	7.95E-03	1.19E-02	7.53E-03	1.08E-02	0	0	0	0	1.11E-03	3.52E-03

4 结束语

灰狼优化算法是一种模拟自然界中灰狼群体社会等级机制和捕食行为的新型群体智能优化方法。本文利用反向学习策略产生初始灰狼个体位置以维持群体多样性;受 PSO 算法的启发,使收敛因子随进化迭代次数增加而非线性动态变化,以协调算法的勘探和开采能力;对当前最优灰狼个体进行变异操作以减少算法出现早熟收敛的可能性。提出一种基于非线性收敛因子的改进灰狼优化算法用于解决高维函数优化问题。通过对 10 个标准测试函数进行数值实验,结果表明,改进灰狼优化算法具有较强的寻优性能。

参考文献:

- [1] 王冰. 基于局部最优解的改进人工蜂群算法[J]. 计算机应用研究, 2014, 31(4): 1023-1026.
- [2] 井福荣, 郭肇禄, 罗会兰, 等. 应用精英反向学习的引力搜索算法[J]. 计算机应用研究, 2015, 32(12): 3638-3641.
- [3] Leung Y, Wang Yuping. An orthogonal genetic algorithm with quantization for global numerical optimization[J]. *IEEE Trans on Evolutionary Computation*, 2001, 5(1): 41-53.
- [4] Coelho L, Ayala H, Mariani V. A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization[J]. *Applied Mathematics and Computation*, 2014, 234(5): 452-459.
- [5] Akay B, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization [J]. *Information Sciences*, 2012, 192(1): 120-142.
- [6] Sun Yang, Zhang Lingbo, Gu Xingsheng. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization[J]. *Neurocomputing*, 2012, 98(3): 76-89.
- [7] Gandomi A H, Yang Xinshe, Talatahari S, et al. Firefly algorithm with chaos[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2013, 18(2): 89-98.
- [8] Mirjalili S, Lewis A. Adaptive gbest-guided gravitational search algorithm[J]. *Neural Computing and Applications*, 2014, 25(7): 1569-1584.
- [9] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimization[J]. *Advances in Engineering Software*, 2014, 69(7): 46-61.
- [10] Song H M, Sulaiman M, Mohamed M. An application of grey wolf optimizer for solving combined economic emission dispatch problems[J].

International Review on Modeling and Simulations, 2014, 7(5): 838-844.

- [11] 杨红光, 刘建生. 一种结合灰狼优化和 K-均值的混合聚类算法[J]. 江西理工大学学报, 2015, 36(5): 85-89.
- [12] El-Gaafary A, Mohamed Y, Hemeida A, et al. Grey wolf optimization for multi input multi output system[J]. *Universal Journal of Communications and Networks*, 2015, 3(1): 1-6.
- [13] Song Xianhai, Tang Li, Zhao Suta, et al. Grey wolf optimizer for parameter estimation in surface waves[J]. *Soil Dynamics and Earthquake Engineering*, 2015, 75(5): 147-157.
- [14] Madadi A, Motlagh M. Optimal control of DC motor using grey wolf optimizer algorithm[J]. *Technical Journal of Engineering and Applied Science*, 2014, 4(4): 373-379.
- [15] Komaki G, 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(3): 109-120.
- [16] Mirjalili S. How effective is the grey wolf optimizer in training multi-layer perceptrons[J]. *Applied Intelligence*, 2015, 42(2): 608-619.
- [17] Zhu Aijun, Xu Chuanpei, Li Zhi, 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.
- [18] 龙文, 赵东泉, 徐松金. 求解约束优化问题的改进灰狼优化算法[J]. 计算机应用, 2015, 35(9): 2590-2595.
- [19] Saremi S, Mirjalili S Z, Mirjalili S M. Evolutionary population dynamics and grey wolf optimizer[J]. *Neural Computing and Applications*, 2015, 26(5): 1257-1263.
- [20] Haupt R, Haupt S. *Practical genetic algorithm*[M]. New York: Wiley, 2004.
- [21] Tizhoosh H. Opposition-based learn: a new scheme for machine intelligence[C]//Proc of International Conference on Computational Intelligence for Modeling Control and Automation. [S. l.]: IEEE Press, 2005: 695-701.
- [22] Wang Yong, Cai Zixing, Zhou Yuren, et al. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique[J]. *Structural and Multidisciplinary optimization*, 2009, 37(1): 395-413.
- [23] Gao Weifeng, Liu Sanyang, Huang Lingling. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning[J]. *IEEE Trans on Cybernetics*, 2013, 43(3): 1011-1024.

(上接第 页)

- [3] 王国霞, 刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用, 2012, 48(7): 66-76.
- [4] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. *Communications of the ACM*, 1992, 35(12): 61-70.
- [5] Karypis G. Evaluation of item-based top-n recommendation algorithms[C]//Proc of the 10th International Conference on Information and Knowledge Management. 2001.
- [6] Hariri N, Mobasher B, Burke R. Context adaptation in interactive recommender systems[C]//Proc of ACM Conference on Recommender Systems. 2014: 41-48.
- [7] Sedhain S, Sanner S, Brazhuanas D, et al. Social collaborative filtering for cold-start recommendations[C]//Proc of ACM Conference on Recommender Systems. 2014: 345-348.
- [8] Vermaas R, Vandic D, Frasincar F. Incremental cosine computations

for search and exploration of tag spaces[M]//Database and Expert Systems Applications. Berlin Heidelberg: Springer, 2012: 156-167.

- [9] Begelman G, Keller P, PSmadja F. Automated tag clustering: improving search and exploration in the tag space[C]//Proc of the 15th Collaborative Web Tagging Workshop, International World Wide Web Conference. 2006.
- [10] Lemire D, Maclachlan A. Slope one predictors for online rating-based collaborative filtering[C]//SIAM; Proc of SIAM Data Mining Conference. 2005: 21-23.
- [11] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender system[J]. *ACM Trans on Information Systems*, 2004, 22(1): 5-53.
- [12] 罗琦, 缪昕杰, 魏倩. 稀疏数据集协同过滤算法的进一步研究[J]. 计算机科学, 2014, 41(6): 264-268.
- [13] 冷亚军, 陆青, 张俊岭. 结合类别偏好信息的 item-based 协同过滤算法[J]. 计算机应用研究, 2015, 33(3): 669-672.