

协调探索和开发能力的改进灰狼优化算法

龙 文^{1,3}, 伍铁斌^{2†}

(1. 贵州财经大学 贵州省经济系统仿真重点实验室, 贵阳 550025; 2. 湖南人文科技学院
能源与机电工程学院, 湖南 娄底 417000; 3. 贵州财经大学 数学与统计学院, 贵阳 550025)

摘 要: 提出一种协调探索和开发能力的灰狼优化算法. 利用佳点集方法初始化灰狼个体的位置, 为全局搜索多样性奠定基础; 为协调算法的全局探索和局部开发能力, 给出一种基于正切三角函数描述的非线性动态变化控制参数; 为加快算法的收敛速度, 受粒子群优化算法个体记忆功能的启发, 设计一种新的个体位置更新公式. 10 个标准函数的测试结果表明, 改进灰狼优化 (IGWO) 算法能够有效地协调其对问题搜索空间的探索和开发能力.

关键词: 灰狼优化算法; 探索能力; 开发能力; 非线性控制参数; 佳点集方法

中图分类号: TP301.6

文献标志码: A

Improved grey wolf optimization algorithm coordinating the ability of exploration and exploitation

LONG Wen^{1,3}, WU Tie-bin^{2†}

(1. Guizhou Key Laboratory of Economics System Simulation, Guizhou University of Finance and Economics, Guiyang 550025, China; 2. School of Energy and Electrical Engineering, Hunan University of Humanities Science and Technology, Loudi 417000, China; 3. School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang 550025, China)

Abstract: An improved grey wolf optimization (IGWO) algorithm is proposed to solve global continuous optimization problems. The good point set method is used to initiate the grey wolves individuals' position, which strengthens the diversity of initial individuals in the global searching process. A nonlinear strategy based on the tangent trigonometric function for updating the control parameter is given to balance the exploration and exploitation abilities of the proposed algorithm. Inspired by the particle swarm optimization (PSO) algorithm, a new position update equation of individuals by incorporating the information of individual historical best solution into the position update equation is designed to speed up convergence. The experimental results and comparisons with the classical GWO algorithm and other improved GWO algorithms using a set of well-known benchmark test functions show that the proposed IGWO algorithm can balance the exploration and exploitation to the problem's solution space effectively during evolution.

Keywords: grey wolf optimization algorithm; exploration ability; exploitation ability; nonlinear control parameter; good point set method

0 引 言

模拟自然界中灰狼群体捕猎行为和分工等级机制, Mirjalili 等^[1]提出了一种新型群体智能算法, 即灰狼优化算法 (GWO). GWO 算法具有模型简单、较少参数设置、寻优性能较好等优点, 且研究结果表明其性能明显优于粒子群优化 (PSO) 算法和引力搜索算法 (GSA)^[1]. 因此, GWO 算法在多输入多输出电力系

统^[2]、经济调度指派问题^[3]、多传感器训练^[4]、面波参数优化^[5]、流水车间调度问题^[6]、电力潮流优化^[7]、 K -均值聚类优化^[8]、医学图像融合优化^[9]、多目标宫颈癌病变分类^[10]等领域中有着广泛应用.

基本 GWO 算法存在探索和开发能力难以协调、求解精度低等缺点. 为了改善基本 GWO 算法的性能, 文献 [11] 采用 5 种非线性曲线调整控制参数, 引入变

收稿日期: 2016-12-04; 修回日期: 2017-03-10.

基金项目: 国家自然科学基金项目 (61463009); 商务部与贵州财经大学联合基金项目 (2016SWBZD13); 贵州省科学技术基金项目 (黔科合基础 [2016]1022); 湖南省自然科学基金项目 (2016JJ3079); 湖南省教育厅青年基金项目 (14B097).

作者简介: 龙文 (1977—), 男, 教授, 博士生导师, 从事机器学习、智能计算、复杂系统建模与控制等研究; 伍铁斌 (1981—), 男, 副教授, 博士, 从事智能优化算法、复杂系统建模、优化与控制等研究.

†通讯作者. E-mail: wutiebin81@163.com

异策略对个体位置进行更新,提出了一种控制参数非线性调整的GWO算法;文献[12]随机选择3个个体与群体决策层个体进行算术交叉操作,对决策层个体采用多样性变异策略以避免出现早熟收敛;文献[13]结合GWO和DE的优点,提出了一种混合算法用于求解全局函数优化和3维芯片测试;文献[14]利用Tent混沌映射和对立学习初始化改进位置更新方程,提出了一种基于混沌局部搜索的改进GWO算法,用于优化在线序贯极限学习机模型的参数;文献[15]提出了一种灰狼优化模式搜索算法,用于优化停电风险预防;文献[16]结合混沌初始化、精英反向学习和混沌扰动,提出了一种混合GWO算法,用于处理高维优化问题;文献[17]提出了一种基于非线性控制参数策略的改进GWO算法,用于求解全局工程优化问题.

为了有效避免基本GWO算法易出现早熟收敛这一缺陷,本文提出一种协调全局探索和局部开发能力的改进GWO算法.所提出的算法在以下3个方面作了改进:

- 1) 利用佳点集方法生成初始种群,以保证个体能够尽可能地均匀分布在搜索空间中;
- 2) 给出一种基于三角函数的非线性控制参数,以平衡GWO算法的全局探索和局部开发能力;
- 3) 受PSO算法的启发,基于个体记忆功能设计出一种新的灰狼个体位置更新公式,以加快算法的收敛速度.

标准函数测试结果表明,改进GWO算法能够有效地协调其探索和开发能力.

1 基本灰狼优化算法

在利用GWO算法求解优化问题时,假设在 D 维搜索空间中,第 i 只灰狼个体的位置为 $X_i = (X_i^1, X_i^2, \dots, X_i^D)$, $i = 1, 2, \dots, N$, N 为种群规模.全局最优解为 α ,次最优解为 β ,第3最优解为 δ ,其他个体为 ω .

群体接近并包围猎物行为的数学模型为

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

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 定义为

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

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

rand_1 和 rand_2 为 $[0, 1]$ 间的随机数; a 为控制参数,随迭代次数的增加从2线性减小到0,即

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

t 为当前迭代次数, t_{\max} 为最大迭代次数.

群体中其他灰狼个体 X_i 根据 α 、 β 和 δ 的位置 X_α 、 X_β 和 X_δ 更新各自的位置,有

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

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

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

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

$f(X_j(t))$ 表示第 j 个个体在第 t 代的适应度值.

基本GWO算法流程如图1所示.

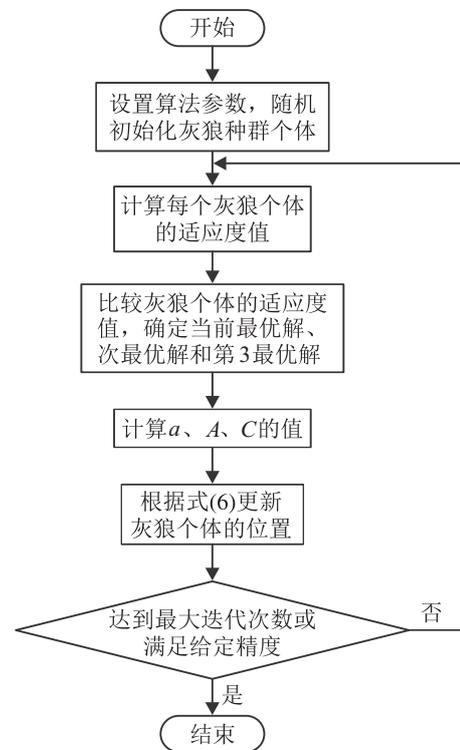


图1 基本GWO算法流程

2 改进灰狼优化算法

2.1 佳点集种群初始化

在进行迭代前,基本GWO算法的初始种群个体是随机产生的,难以保证初始种群的多样性,从而在一定程度上影响算法的搜索效率.

佳点集是一种有效的、能够均匀选点的方法,与随机方法相比,利用佳点集方法取点能够更均匀地分布在搜索空间中^[18].目前,佳点集方法已在GA、PSO、DE、ABC等群体智能优化算法中得到了成功的应用,因此,本文将佳点集方法应用到GWO算法中生成初始种群个体.利用佳点集产生初始种群的具体原理详见文献[19].

2.2 非线性控制参数策略

对于群体智能优化算法而言,如何协调其全局探索能力和局部开发能力至关重要. 全局探索意味着群体需要探测更广泛的搜索区域,避免算法陷入局部最优,与群体算法的鲁棒性息息相关;局部开发则主要强调充分利用已有的信息对群体的某些区域进行精细搜索,对算法的收敛速度具有决定性的影响. 毫无疑问,只有当群体智能优化算法能够较好地协调其全局探索和局部开发能力时,才能具备较强的鲁棒性和较快的收敛速度. 因此,作为一种新型群体智能优化算法,GWO能否获得较好寻优性能的关键是如何在全局探索和局部开发能力之间进行有效的协调.

由第1节可知,GWO算法主要由猎物定位和灰狼个体移动两个步骤组成. 根据式(1),参数 A 对平衡GWO算法的全局探索和局部开发能力具有关键的作用. 当 $|A| > 1$ 时,群体将扩大搜索范围,以找到更好的候选解,即为GWO算法的全局探索能力;当 $|A| < 1$ 时,群体将缩小搜索范围,在局部区域进行精细搜索,即为GWO算法的局部开发能力. 同时,由式(2)可知,在进化过程中, A 的值随控制参数 a 的变化而不断变化. 换言之,GWO算法的全局探索和局部开发能力在很大程度上依赖于控制参数 a . 但是,由式(4)可知,控制参数 a 随迭代次数的增加由2线性减小到0.

然而,GWO算法的寻优搜索过程非常复杂,控制参数 a 随迭代次数线性递减的策略不能完全体现出实际的优化搜索过程. 文献[11]和文献[17]提出了控制参数 a 随进化迭代次数非线性变化,标准函数测试结果表明,采用非线性变化策略比线性策略获得了更好的寻优性能. 受PSO算法中惯性权重设置的启发,本文提出一种基于正切三角函数的非线性控制参数策略,即

$$a(t) = a_{\text{initial}} - (a_{\text{initial}} - a_{\text{final}}) \times \tan\left(\frac{1}{\varepsilon} \cdot \frac{t}{t_{\text{max}}}\pi\right). \quad (8)$$

其中: a_{initial} 和 a_{final} 分别为控制参数 a 的初始值和最终值; $\varepsilon > 0$ 为非线性调节系数; t 为当前迭代次数, t_{max} 为最大迭代次数. 由式(8)可知,控制参数 a 随迭代次数的增加非线性动态变化,以协调GWO算法的全局探索和局部开发能力.

2.3 基于个体记忆功能的位置更新公式

分析位置更新式(5)和(6)可知,GWO算法在位置迭代更新过程中只考虑了个体当前位置信息和群体历史最优位置信息,并实现了狼群信息的交流,最终通过个体在搜索空间移动收敛到最优解. 但GWO

算法未考虑个体自身的经验,说明其是一种缺乏个体自身位置记忆性的算法.

为了改善GWO算法的局部开发能力,受粒子群优化(PSO)算法的启发,将PSO算法中对粒子自身运动历史最优解进行记忆保存的思想引入到GWO算法中,对个体的记忆功能加以改进,使其能够记忆自身进化过程中的最优解. 为此,本文提出一种基于个体自身记忆功能的位置更新公式替代式(6),表示为

$$X_i^d(t+1) = b_1 \cdot \sum_{j=\alpha,\beta,\delta} w_j X_{i,j}^d(t+1) + b_2 \cdot \text{rand}_3 \cdot (P_{\text{best}}^d - X_i^d(t)). \quad (9)$$

其中: $b_1 \in [0, 1]$ 为群体交流系数, $b_2 \in [0, 1]$ 为个体记忆系数, rand_3 为 $[0,1]$ 间的随机数, P_{best}^d 为第 i 只灰狼个体所经历过的最佳位置. 类似于PSO算法,通过调节 b_1 和 b_2 的值,可以协调群体交流与个体记忆对算法搜索的影响.

2.4 算法步骤

综上所述,改进灰狼优化(IGWO)算法的步骤如下所示.

算法1 IGWO算法.

Step 1: 设置算法参数: 种群规模 N , 最大迭代次数 t_{max} , 参数 a 的初始值 a_{initial} , 最终值 a_{final} , 非线性调节系数 ε , 群体交流系数 b_1 和个体记忆系数 b_2 ;

Step 2: 利用2.1节所描述的佳点集方法产生初始化灰狼种群 $\{X_i, i = 1, 2, \dots, N\}$, 令 $t = 0$;

Step 3: 计算每个个体的适应度值 $\{f(X_i), i = 1, 2, \dots, N\}$ 并排序, 当前适应度值排列前3位的个体分别记为 α 、 β 和 δ , 其对应的位置分别为 X_α 、 X_β 和 X_δ ;

Step 4: 重复Step 3, 直至达到最大迭代次数或满足给定精度.

IGWO算法步骤的伪代码表示如下:

```
while (t < t_max) do
  for i = 1 to N do
    根据式(8)计算控制参数 $\alpha$ 的值
    根据式(2)和(3)更新参数 $A$ 和 $C$ 的值
    根据式(9)更新其他个体的位置
  end for
  计算每个个体的适应度 $\{f(X_i), i = 1, 2, \dots, N\}$ 
  更新 $\alpha$ 、 $\beta$ 、 $\delta$ 及其相应位置 $X_\alpha$ 、 $X_\beta$ 和 $X_\delta$ 
  t = t + 1
end while
```

IGWO算法的时间复杂度计算如下: 计算群体中每个个体适应度值的时间复杂度为 $O(N)$, N 为种群

规模;个体位置更新操作的时间复杂度为 $O(N^2+N)$;群体循环迭代的时间计算复杂度为 $O(N^2)$;佳点集初始化操作的时间复杂度忽略不计.因此,IGWO算法总的复杂度为 $O(N^2)$.

3 数值实验及比较

为了验证IGWO算法求解复杂高维优化问题的能力,本节将其用于优化求解10个国际上通用的标准测试函数,与基本GWO算法、Hybrid DE and GWO

(记为DE-GWO)算法^[13]、混合GWO(记为HGWO)算法^[16]和Modified GWO(记为mGWO)算法^[17]进行性能比较,并分析重要参数对IGWO算法性能的影响.

3.1 测试函数及性能指标

为了提供一个充分且全面的比较实验,从文献[3]中选取10个标准测试函数作为实验对象,10个标准测试函数的函数名、表达式、搜索区间、理论最优值和收敛精度如表1所示,其中 f_{\min} 为理论最优值.

表1 标准测试函数

函数名	表达式	搜索区间	f_{\min}	收敛精度
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	$[-100, 100]$	0	1×10^{-08}
Schwefel 2.22	$f_2(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	$[-10, 10]$	0	1×10^{-08}
Schwefel 1.2	$f_3(x) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0	1×10^{-08}
Schwefel 2.1	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq d\}$	$[-100, 100]$	0	1×10^{-08}
Rosenbrock	$f_5(x) = \sum_{i=1}^d [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-5, 10]$	0	1×10^{-0}
Step	$f_6(x) = \sum_{i=1}^d (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]$	0	1×10^{-08}
Quartic	$f_7(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0	1×10^{-04}
Rastrigin	$f_8(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0	1×10^{-08}
Ackley	$f_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0	1×10^{-08}
Griewank	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0	1×10^{-08}

本文选取2个性能指标来评价算法的性能^[20].第1个性能指标是结果精确度(AC),该指标反映了算法所得结果与问题全局最优解的接近程度.假设一个问题的全局最优解是 X_{opt} , t 次迭代后算法所找到的最优解为 P_{best} ,则所求结果的精确度定义为

$$AC = |f(P_{\text{best}}) - f(X_{\text{opt}})|. \quad (10)$$

第2个性能指标是成功率(SR),即多次实验中算法收敛到问题最优解的比例,定义为

$$SR = \frac{z'}{z} \times 100\%. \quad (11)$$

其中: z 为总的实验次数, z' 为算法收敛到问题最优解的实验次数.在实验中,如果算法迭代所获得结果对应的AC值小于设定的收敛精度(见表1中最后一列),则认为该次实验成功收敛到问题的最优解.

3.2 与基本GWO算法的比较

采用IGWO算法对10个标准测试函数进行求解,并与基本GWO算法的性能进行比较,通过统计两种

算法不同性能测度的结果,对比分析两种算法的优化性能,验证IGWO算法的有效性.在对比实验中,为了进行公平的比较,两种算法均使用相同的实验参数,即最大适应度函数评价次数为15000(种群规模 $N = 30$,最大迭代次数 $t_{\text{max}} = 500$),控制参数 a 的初始值 $a_{\text{initial}} = 1$ 、最终值 $a_{\text{final}} = 0$,非线性调节系数 $\varepsilon = 5$,群体交流系数 $b_1 = 0.5$,个体记忆系数 $b_2 = 0.5$.对于每个测试函数,基本GWO算法和IGWO算法均独立运行30次,记录其平均精度、精度标准差和成功率,结果如表2和表3所示.

由表2可知:无论是低维($D = 30$),还是高维($D = 100, 500, 1000$),IGWO算法在5个函数($f_1, f_3, f_6, f_8, f_{10}$)上的30次实验均能够收敛到理论最优值0;对于函数 f_2 和 f_4 ,IGWO算法获得的结果虽然没能收敛到0,但非常接近于0.另外,IGWO算法的标准差值非常小,说明其具有较强的鲁棒性.与基本GWO算法相比,除了函数 f_5 和 f_6 ,IGWO算法在其他

8个函数上均得到了较好的平均值和标准差. 在寻优成功率方面, 基本GWO算法在高维函数上的寻优成功率基本为0, 说明其处理高维问题的能力有待加强; 除了函数 f_5 , IGWO算法在其他函数上的寻优成功率

均为100%. 为了更直观地反映IGWO算法的收敛性能, 图2给出了两种算法对部分测试函数的寻优收敛曲线. 从图2可以看出, 与基本GWO算法相比, IGWO算法具有较快的收敛速度和较高的寻优精度.

表2 IGWO和GWO的寻优结果比较

函数	算法	$D = 30$ 维		$D = 100$ 维		$D = 500$ 维		$D = 1000$ 维	
		平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差
f_1	GWO	3.30e-028	2.32e-028	9.77e-013	1.11e-012	1.25e-003	5.22e-004	2.38e-001	6.33e-002
	IGWO	0	0	0	0	0	0	0	0
f_2	GWO	1.16e-016	6.75e-017	4.16e-008	9.73e-009	1.10e-002	9.49e-004	6.30e-002	4.16e-001
	IGWO	0	0	5.67e-308	0	2.89e-302	0	3.56e-300	0
f_3	GWO	2.99e-006	3.01e-006	2.53e+002	2.34e+002	3.40e+005	1.16e+005	2.71e+006	1.80e+006
	IGWO	0	0	0	0	0	0	0	0
f_4	GWO	7.01e-007	9.00e-007	1.19e+000	9.87e-001	6.83e+001	5.77e+000	7.76e+001	3.78e+002
	IGWO	2.03e-307	0	1.48e-294	0	8.50e-288	0	7.73e-286	0
f_5	GWO	2.73e+001	5.46e-001	9.78e+001	6.26e-001	4.98e+002	1.76e-001	1.06e+003	2.18e+001
	IGWO	2.89e+001	1.36e-002	9.89e+001	2.46e-002	4.99e+002	1.83e-002	9.99e+002	1.57e-002
f_6	GWO	0	0	0	0	1.40e+000	1.26e+000	1.16e+001	5.78e+000
	IGWO	0	0	0	0	0	0	0	0
f_7	GWO	2.06e-003	1.15e-003	6.15e-003	2.03e-003	4.97e-002	1.60e-002	5.89e-002	1.71e-002
	IGWO	4.10e-005	3.30e-005	5.05e-005	5.76e-005	6.34e-005	1.13e-004	1.03e-004	6.79e-004
f_8	GWO	3.57e+000	3.74e+000	1.00e+001	6.90e+000	7.64e+001	3.03e+001	7.71e+001	2.71e+001
	IGWO	0	0	0	0	0	0	0	0
f_9	GWO	1.11e-013	2.59e-014	1.17e-007	4.31e-008	2.07e-003	4.16e-004	2.18e-003	4.69e-004
	IGWO	8.88e-016	0	1.95e-015	1.72e-015	3.37e-015	1.82e-015	3.73e-015	1.90e-0150
f_{10}	GWO	7.91e-003	1.28e-002	1.02e-002	1.36e-002	1.97e-002	4.12e-002	1.99e-002	4.71e-002
	IGWO	0	0	0	0	0	0	0	0

表3 GWO算法和IGWO算法的寻优成功率比较 %

函数	算法	$D = 30$	$D = 100$	$D = 500$	$D = 1000$
f_1	GWO	100	100	0	0
	IGWO	100	100	100	100
f_2	GWO	100	100	0	0
	IGWO	100	100	100	100
f_3	GWO	20	0	0	0
	IGWO	100	100	100	100
f_4	GWO	10	0	0	0
	IGWO	100	100	100	100
f_5	GWO	0	0	0	0
	IGWO	0	0	0	0
f_6	GWO	100	100	30	0
	IGWO	100	100	100	100
f_7	GWO	20	0	0	0
	IGWO	100	100	100	100
f_8	GWO	80	0	0	0
	IGWO	100	100	100	100
f_9	GWO	100	60	0	0
	IGWO	100	100	100	100
f_{10}	GWO	60	60	0	0
	IGWO	100	100	100	100

3.3 与改进GWO算法的比较

为了进一步验证IGWO算法的有效性, 将其与其他3种性能较好的改进GWO算法 (DE-GWO算法^[13]、mGWO算法^[17]和HGWO算法^[16]) 的结果进行比较. 3种算法的参数设置如下: 在DE-GWO算法中, 收缩因子 $F = 0.5$, 交叉概率 $P_c = 0.2$, 最大函数评价次数为15000 (即种群规模 $N = 30$, 最大迭代次数 $t_{max} = 500$); 在mGWO算法中, 最大函数评价次数为90000 (即种群规模 $N = 30$, 最大迭代次数 $t_{max} = 3000$); 在HGWO算法中, 最大函数评价次数为50000 (即种群规模 $N = 50$, 最大迭代次数 $t_{max} = 1000$). 为了确保比较的公平性, DE-GWO、mGWO和HGWO算法的结果直接来源于各自参考文献. 表4和表5分别给出了几种算法的结果比较, 其中黑色粗体表示比较算法中的最好结果.

由表4可知, 总体而言, 在3种比较算法中, IGWO算法的寻优性能明显优于其他两种算法, DE-GWO算法的性能最差. 与DE-GWO算法相比, 除了函数 f_5 , IGWO算法在其他9个函数上均获得了较好的

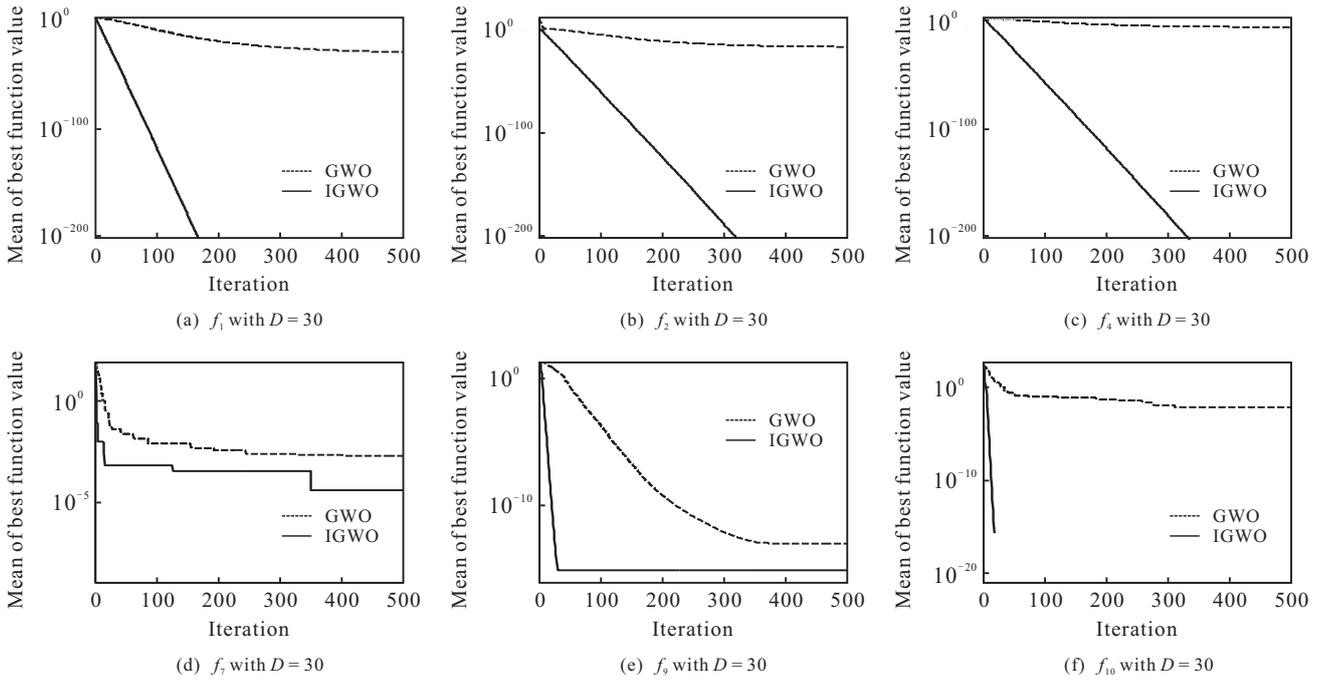


图2 GWO和IGWO算法的寻优收敛曲线

寻优性能;对于函数 f_5 , DE-GWO算法则取得了稍优的结果. 与mGWO算法相比, IGWO算法在7个函数($f_1, f_2, f_3, f_4, f_6, f_7, f_9$)上获得了较好的平均值和标准差;对于函数 f_8 和 f_{10} ,两种算法得到了相似的测试结果,但IGWO算法的最大适应度函数评价次数仅为mGWO算法的六分之一;对于函数 f_5 , mGWO算法则取得了稍好的结果.

表4 IGWO、DE-GWO和mGWO的结果比较($D = 30$)

函数	统计结果	DE-GWO	mGWO	IGWO
f_1	平均精度	1.12e-032	6.44e-205	0
	标准差	2.32e-032	0	0
f_2	平均精度	9.33e-020	3.34e-119	0
	标准差	6.92e-020	4.95e-119	0
f_3	平均精度	3.18e-008	2.74e-052	0
	标准差	6.55e-008	1.19e-051	0
f_4	平均精度	4.17e-008	1.20e-051	2.03e-307
	标准差	4.56e-008	4.25e-051	0
f_5	平均精度	2.65e+001	2.69e+001	2.89e+001
	标准差	7.03e-001	8.52e-001	1.36e-002
f_6	平均精度	0.378 23	0.786 295 4	0
	标准差	0.226 26	0.244 928 7	0
f_7	平均精度	1.49e-003	2.61e-004	4.10e-005
	标准差	7.54e-004	1.76e-004	3.30e-005
f_8	平均精度	0.227 48	0	0
	标准差	0.920 19	0	0
f_9	平均精度	4.27e-014	7.82e-015	8.88e-016
	标准差	4.37e-015	7.94e-016	0
f_{10}	平均精度	1.37e-003	0	0
	标准差	5.82e-003	0	0

表5 IGWO和HGWO的结果比较

函数	维数	HGWO		IGWO	
		平均精度	标准差	平均精度	标准差
f_1	100	2.66e-035	1.09e-034	0	0
	500	3.07e-014	1.51e-014	0	0
	1000	4.62e-010	2.73e-010	0	0
f_2	100	7.51e-021	6.05e-021	5.67e-308	0
	500	5.61e-009	1.78e-009	2.89e-302	0
	1000	1.81e-005	1.00e-005	3.56e-300	0
f_4	100	3.54e-005	2.50e-005	1.48e-294	0
	500	1.81e-003	2.09e-003	8.50e-288	0
	1000	5.77e-001	7.03e-002	7.73e-286	0
f_5	100	6.14e+000	7.00e-001	9.89e+001	2.46e-002
	500	9.25e+001	5.22e-001	4.99e+002	1.83e-002
	1000	9.96e+002	0	9.99e+002	1.57e-002
f_6	100	1.03e-005	2.18e-006	0	0
	500	3.76e-002	5.87e-003	0	0
	1000	8.49e+00	2.75e+00	0	0
f_7	100	1.77e-008	2.33e-008	5.05e-005	5.76e-005
	500	4.21e-006	1.90e-006	6.34e-005	1.13e-004
	1000	6.17e-003	3.53e-003	1.03e-004	6.79e-004
f_8	100	1.13e-013	0	0	0
	500	1.01e-010	1.18e-010	0	0
	1000	3.95e-007	2.52e-007	0	0
f_9	100	7.55e-014	0	1.95e-015	1.72e-015
	500	7.62e-009	2.50e-010	3.37e-015	1.82e-015
	1000	5.45e-007	7.08e-008	3.73e-015	1.90e-015
f_{10}	100	0	0	0	0
	500	5.53e-015	1.92e-015	0	0
	1000	2.59e-011	1.16e-011	0	0

由表5可知,与HGWO算法相比, IGWO算法在7个函数($f_1, f_2, f_4, f_6, f_8, f_9, f_{10}$)不同维数上均能获得

表8 不同的 ε 值对IGWO算法性能的影响结果比较($D = 30$ 维)

函数	$\varepsilon = 1$		$\varepsilon = 5$		$\varepsilon = 25$		$\varepsilon = 50$		$\varepsilon = 100$	
	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差
f_1	0	0	0	0	0	0	0	0	0	0
f_2	6.19e-228	0	0	0	8.45e-307	0	2.03e-305	0	2.83e-304	0
f_3	0	0	0	0	0	0	0	0	0	0
f_4	7.87e-218	0	2.03e-307	0	1.27e-294	0	2.64e-292	0	3.66e-292	0
f_5	2.86e+001	3.46e-001	2.89e+001	1.36e-002	2.90e+001	2.03e-02	2.90e+001	5.89e-002	2.90e+001	8.69e-002
f_6	0	0	0	0	0	0	0	0	0	0
f_7	7.48e-005	6.58e-005	4.10e-005	3.30e-005	7.85e-005	7.14e-005	1.27e-004	8.34e-005	7.18e-005	6.86e-005
f_8	0	0	0	0	0	0	0	0	0	0
f_9	8.88e-016	0	8.88e-016	0	8.88e-016	0	8.88e-016	0	8.88e-016	0
f_{10}	0	0	0	0	0	0	0	0	0	0

从表7的结果可以看出,当参数 a_{initial} 和 a_{final} 选取不同的值时,除了函数 f_2 和 f_4 ,IGWO算法在其余8个测试函数上获得的结果相差不是很大,这充分说明IGWO算法对参数 a_{initial} 和 a_{final} 不太敏感.但总体而言,当 $a_{\text{initial}} = 1, a_{\text{final}} = 0$ 时,IGWO算法的总体寻优性能最好.因此, $a_{\text{initial}} = 1, a_{\text{final}} = 0$ 是合理的参数选择.

由表8可知,当参数 ε 选取不同的值时,除了函数 f_2 、 f_4 和 f_5 ,IGWO算法在其他7个测试函数上得到的结果变化很小,这说明IGWO算法对参数 ε 不太敏

感.当 $\varepsilon = 5$ 时,IGWO算法的总体寻优性能最佳,因此 $\varepsilon = 5$ 是最佳的参数选择.

3.6 b_1 和 b_2 对IGWO算法的性能影响分析

在如式(9)所示的个体位置更新公式中,群体交流参数 b_1 和个体记忆系数 b_2 对其影响较大,通过调节 b_1 和 b_2 的值,使得算法具有不同的探索和开发能力.本小节通过选取不同的 b_1 和 b_2 值分析其对IGWO算法性能的影响,算法中其他参数设置与3.2节相同.表9给出了IGWO算法选取不同 b_1 和 b_2 值时的结果比较,黑粗体表示比较算法中的最好结果.

表9 不同的 b_1 和 b_2 值对IGWO算法性能的影响结果比较($D = 30$ 维)

函数	$b_1 = 0.1$ 和 $b_2 = 0.9$		$b_1 = 0.3$ 和 $b_2 = 0.7$		$b_1 = 0.5$ 和 $b_2 = 0.5$		$b_1 = 0.7$ 和 $b_2 = 0.3$		$b_1 = 0.9$ 和 $b_2 = 0.1$	
	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差
f_1	0	0	0	0	0	0	0	0	0	0
f_2	1.28e-192	0	6.42e-232	0	1.20e-282	0	0	0	4.66e-272	0
f_3	0	0	0	0	0	0	0	0	0	0
f_4	1.58e-191	0	1.77e-229	0	6.96e-279	0	2.03e-307	0	2.20e-246	0
f_5	2.89e+001	3.94e-002	2.90e+001	1.91e-002	2.90e+001	1.68e-002	2.89e+001	1.36e-002	2.89e+001	6.32e-002
f_6	0	0	0	0	0	0	0	0	0	0
f_7	5.12e-005	4.65e-005	5.20e-005	6.76e-005	7.36e-005	6.63e-005	4.10e-005	3.30e-005	8.81e-005	5.41e-005
f_8	0	0	0	0	0	0	0	0	0	0
f_9	8.88e-016	0	8.88e-016	0	8.88e-016	0	8.88e-016	0	2.66e-015	1.87e-015
f_{10}	0	0	0	0	0	0	0	0	0	0

由表9的比较结果可知,当 b_1 和 b_2 取不同的值时,除了函数 f_2 和 f_4 ,IGWO算法对其他8个函数的测试结果相差不大,这说明IGWO算法对参数 b_1 和 b_2 值不太敏感.从5组不同的取值结果来看,当 $b_1 = 0.7, b_2 = 0.3$ 时,IGWO算法得到的结果相对较好.因此, $b_1 = 0.7, b_2 = 0.3$ 是合理的参数取值.

4 结论

本文提出了一种基于控制参数非线性变化和修改个体位置更新公式的改进GWO算法.所提算法利用佳点集方法生成初始种群以维持初始群体的多样性;引入正切函数描述控制参数使其随迭代次数

增加非线性动态变化,以协调GWO算法的全局探索和局部开发能力;受PSO算法启发,设计出一种基于个体记忆功能的改进个体位置更新公式.通过对10个测试函数进行数值实验表明,与GWO、DE-GWO、mGWO和HGWO算法相比,本文提出的IGWO算法具有较强的竞争力和较小的适应度函数评价次数.

参考文献(References)

[1] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(7): 46-61.
 [2] El-Gaafary A, Mohamed Y, Hemeida A, et al. Grey

- wolf optimization for multi input multi output system[J]. Universal J of Communications and Networks, 2015, 3(1): 1-6.
- [3] Song H, Sulaiman M, Mohamed M. An application of grey wolf optimizer for solving combined economic emission dispatch problems[J]. Int Review on Modeling and Simulation, 2014, 7(5): 838-844.
- [4] Mirjalili S. How effective is the grey wolf optimizer in training multilayer perceptrons[J]. Applied Intelligent, 2015, 42(2): 608-619.
- [5] Song X, Tang L, Zhao S, et al. Grey wolf optimizer for parameter estimation in surface waves[J]. Soil Dynamics and Earthquake Engineering, 2015, 75(5): 147-157.
- [6] Komaki G, Kayvanfar V. Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time[J]. J of Computational Science, 2015, 8(3): 109-120.
- [7] El-Fergany A, Hasanien H. Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms[J]. Electric Power Components and Systems, 2015, 43(13): 1548-1559.
- [8] 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. (Yang H G, Liu J S. A hybrid clustering algorithm based on grey wolf optimizer and K -means algorithm[J]. J of Jiangxi University of Science and Technology, 2015, 36(5): 85-89.)
- [9] Daniel E, Anitha J, Kamaleshwaran K, et al. Optimum spectrum mask based medical image fusion using grey wolf optimization[J]. Biomedical Signal Processing and Control, 2017, 34(4): 36-43.
- [10] Sahoo A, Chandra S. Multi-objective grey wolf optimizer for improved cervix lesion classification[J]. Applied Soft Computing, 2017, 52(5): 64-80.
- [11] 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. (Wei Z L, Zhao H, Li M D, et al. A grey wolf optimization algorithm based on nonlinear adjustment strategy of control parameter[J]. J of Air Force Engineering University: Natural Science Edition, 2016, 17(3): 68-72.)
- [12] 徐松金, 龙文. 嵌入遗传算子的改进灰狼优化算法[J]. 兰州理工大学学报, 2016, 42(2): 102-108. (Xu S J, Long W. Improved grey wolf optimization algorithm embedded with genetic operators[J]. J of Lanzhou University of Technology, 2016, 42(2): 102-108.)
- [13] Zhu A, Xu C, Li Z, et al. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC[J]. J of Systems Engineering and Electronics, 2015, 26(2): 317-328.
- [14] 徐辰华, 李成县, 王尤军, 等. 基于混沌灰狼优化算法的氧化铝指标预测模型[J]. 广西大学学报: 自然科学版, 2016, 41(6): 1869-1878. (Xu C H, Li C X, Wang Y J, et al. Quality prediction model of alumina sintering process based on chaotic grey wolf optimization algorithm[J]. J of Guangxi University: Natural Science Edition, 2016, 41(6): 1869-1878.)
- [15] Mahdad B, Srairi K. Blackout risk prevention in a smart grid based flexible optimal strategy using grey wolf-pattern search algorithms[J]. Energy Conversion and Management, 2015, 98(6): 411-429.
- [16] 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[J]. 控制与决策, 2016, 31(11): 1991-1997. (Long W, Cai S H, Jiao J J, et al. Hybrid grey wolf optimization algorithm for high-dimensional optimization[J]. Control and Decision, 2016, 31(11): 1991-1997.)
- [17] Mittal N, Singh U, Sohi B S. Modified grey wolf optimizer for global engineering optimization[J]. Applied Computational Intelligence and Soft Computing, 2016(4598): 1-16.
- [18] 张铃, 张钊. 佳点集遗传算法[J]. 计算机学报, 2001, 24(9): 917-922. (Zhang L, Zhang B. Good point set based genetic algorithm[J]. Chinese J of Computers, 2001, 24(9): 917-922.)
- [19] 陈义雄, 梁昔明, 黄亚飞. 基于佳点集构造的改进量子粒子群优化算法[J]. 中南大学学报: 自然科学版, 2013, 44(4): 1409-1414. (Chen Y X, Liang X M, Huang Y F. Improved quantum particle swarm optimization based on good-point set[J]. J of Central South University: Science and Technology, 2013, 44(4): 1409-1414.)
- [20] Engelbrecht A P. Fundamentals of computational swarm intelligence[M]. New York: John Wiley and Sons, 2006.

(责任编辑: 齐 霁)