

基于透镜成像学习策略的灰狼优化算法

龙文¹ 伍铁斌² 唐明珠³ 徐明⁴ 蔡绍洪¹

摘要 在灰狼优化算法中, C 是一个重要的参数, 其功能是负责算法的勘探能力. 目前, 针对参数 C 的研究工作相对较少. 另外, 在算法进化过程中, 群体中其他个体均向 α 、 β 和 δ 所在区域靠近以加快收敛速度. 然而, 算法易陷入局部最优. 为解决以上问题, 本文提出一种改进的灰狼优化算法 (Lens imaging learning grey wolf optimizer algorithm, LIL-GWO). 该算法首先分析了参数 C 的作用, 提出一种新的参数 C 策略以平衡算法的勘探和开采能力; 同时, 分析了灰狼优化算法后期个体均向决策层区域聚集, 从而导致群体多样性较差, 提出一种基于光学透镜成像原理的反向学习策略以避免算法陷入局部最优. 对 LIL-GWO 算法的收敛性进行了证明. 选取 12 个通用的标准测试函数和 30 个 CEC 2014 测试函数进行实验, 在相同的适应度函数评价次数条件下, LIL-GWO 算法在总体性能上优于基本 GWO 算法、改进 GWO 算法和其他比较算法. 最后, 将 LIL-GWO 算法应用于辨识光伏模型的参数, 获得了满意的结果.

关键词 灰狼优化算法, 透镜成像学习, 全局优化, 光伏模型, 参数辨识

引用格式 龙文, 伍铁斌, 唐明珠, 徐明, 蔡绍洪. 基于透镜成像学习策略的灰狼优化算法. 自动化学报, 2020, 46(10): 2148–2164

DOI 10.16383/j.aas.c180695

Grey Wolf Optimizer Algorithm Based on Lens Imaging Learning Strategy

LONG Wen¹ WU Tie-Bin² TANG Ming-Zhu³ XU Ming⁴ CAI Shao-Hong¹

Abstract In the grey wolf optimizer algorithm, C is an important parameter. The exploration capability of GWO mainly depends on the parameter C . At present, few of researchers are aiming at parameter C in GWO algorithm. Many issues need to be settled. In addition, during the evolution process, the other individuals in the population move towards to the α , β and δ which are to accelerate convergence. However, the algorithm is easy to trap in the local optima. In this paper, an improved GWO algorithm called LIL-GWO is proposed to solve these problems. The proposed algorithm firstly analyzes the role of parameter C and presents a new parameter C strategy to balance between exploration and exploitation of GWO. Secondly, at the end of the GWO algorithm, all individuals assemble into the decision-making region which is resulted in poor population multiplicity. A new opposition-learning strategy based on optical lens imaging principle is proposed to help the population jump out of a local optimum. A theoretical proof of convergence for LIL-GWO algorithm is provided. Simulation experiments were conducted on the 12 widely used benchmark test functions and 30 benchmark test functions from the CEC 2014. The overall performance of LIL-GWO algorithm is much better than the basic GWO algorithm, several improved GWO algorithms, and other compared algorithms with the same number of fitness evaluations (FEs). Finally, LIL-GWO is applied to identify the parameters of PV model and the satisfied results are obtained.

Key words Grey wolf optimizer (GWO) algorithm, lens imaging learning (LIL), global optimization, photovoltaic model, parameter identification

Citation Long Wen, Wu Tie-Bin, Tang Ming-Zhu, Xu Ming, Cai Shao-Hong. Grey wolf optimizer algorithm based on lens imaging learning strategy. *Acta Automatica Sinica*, 2020, 46(10): 2148–2164

收稿日期 2018-10-25 录用日期 2019-04-09
Manuscript received October 25, 2018; accepted April 9, 2019
国家自然科学基金 (61463009), 贵州省科学技术基金 (黔科合基础 [2020]1Y012), 湖南省教育厅重点项目 (19A254), 湖南省自然科学基金 (2020JJ4382) 资助
Supported by National Natural Science Foundation of China (61463009), Science and Technology Foundation of Guizhou Province ([2020]1Y012), Key Projects of Education Department of Hunan Province (19A254), and Natural Science Foundation of Hunan Province (2020JJ4382)
本文责任编辑 王鼎
Recommended by Associate Editor WANG Ding
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 娄底 417000 3. 长沙理工大学能源与动力工程学院 长沙 410114 4. 贵州财经大学数学与统计学院 贵阳 550025
1. Key Laboratory of Economics System Simulation, Guizhou

全局优化问题广泛存在于经济、管理、自动控制、工程优化等领域中, 其数学模型可描述为:

$$\text{Minimize } f(\mathbf{X}), \mathbf{X} = [x_1, x_2, \dots, x_D]^T \quad (1)$$

其中, $\mathbf{X} \in R^D$ 为决策向量, $x_i \in [l_i, u_i]$ ($i = 1, 2, \dots, D$), l_i 和 u_i 分别为 x_i 的边界.

University of Finance and Economics, Guiyang 550025 2. Department of Energy and Electrical Engineering, Hunan University of Humanities, Science and Technology, Loudi 417000
3. School of Energy Power and Engineering, Changsha University of Science and Technology, Changsha 410114 4. School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang 550025

对于具有高维、非线性、目标函数不可导等特点的复杂全局优化问题, 传统的优化方法通常不能有效地求解, 而基于群体搜索的智能优化算法对该类问题却获得较好的求解效果. 因此, 基于群体搜索的智能优化算法极大地引起了进化计算领域相关学者的关注和研究. 在过去的 20 余年里, 许多基于群体智能的优化算法如粒子群优化算法^[1]、人工蜂群算法^[2]、花朵授粉算法^[3]、差分进化算法^[4]、布谷鸟搜索算法^[5]、鲸鱼优化算法^[6] 等相继被提出和用于求解复杂全局优化问题.

灰狼优化 (Grey wolf optimizer, GWO) 算法是由澳大利亚学者 Mirjalili 等^[7] 于 2014 年提出的一种新型基于群体智能的优化方法, 它源于对自然界中灰狼群体社会等级制度和捕食行为的模拟, 通过跟踪追捕、包围和攻击等过程, 构建一种有效的优化方案. GWO 算法原理简单、易于编程实现、需调节的参数少, 在电力系统^[8]、自动控制^[9]、图像处理^[10]、能源市场战略招标^[11]、机器学习^[12]、航路规划^[13] 等领域中有着成功的应用.

然而, 与其他群体智能优化算法相似, GWO 算法也存在求解精度低、收敛速度慢和易陷入局部最优等缺点, 在解决复杂高维优化问题时尤甚. 为了克服这些缺陷, 研究学者提出了许多性能较好的改进 GWO 算法, 其大体上分为以下 4 类:

1) 修改控制参数策略. 在 GWO 算法中, \mathbf{A} 和 \mathbf{C} 是两个关键的控制参数, 在平衡算法的勘探和开采能力中起重要作用. 然而, \mathbf{A} 又依赖于另一个参数 \mathbf{a} , 因此, 对参数 \mathbf{a} 的研究较多. Mittal 等^[14] 提出一种基于指数衰减函数的控制参数 \mathbf{a} 策略以增强算法的全局勘探能力. Rodríguez 等^[15] 设计出一种基于模糊逻辑动态自适应调整控制参数 \mathbf{a} 策略以协调算法的勘探和开采能力. 受粒子群优化算法启发, Long 等^[16] 提出一种基于对数衰减函数的非线性控制参数 \mathbf{a} 策略以增强算法的开采能力. 魏政磊等^[17] 比较采用正弦、余弦、正切、对数和二次函数等非线性策略调整控制参数 \mathbf{a} , 结果表明, 采用余弦和二次函数的策略得到较好的性能. 关于控制参数 \mathbf{C} , Rodríguez 等^[15] 提出一种基于模糊逻辑的动态自适应调整策略.

2) 修改位置更新方程. 在 GWO 算法中, 通过决策层个体 (α 、 β 和 δ) 引导进行搜索, 位置更新方程对算法收敛提供了较大的帮助. 然而, 在搜索后期易出现早熟收敛现象. 为了避免算法陷入局部最优, 研究者提出许多修改位置更新方程策略. Long 等^[18] 从群体中随机选择个体与当前个体进行差分操作产生新个体, 从而提高群体的多样性, 减少算法陷入局部最优的概率. Rodríguez 等^[19] 引入平均权重、基于适应度的权重和模糊动态权重修改位置

更新方程, 产生更多有潜力的新个体以增强群体多样性. Jaiswal 等^[20] 引入随机权重系数修改位置更新方程产生新个体, 帮助种群跳出局部最优. Malik 等^[21] 将权重距离系数引入到位置更新方程中, 该修改方程对复杂多峰函数特别有效. 魏政磊等^[22] 根据当前个体适应度与群体平均适应度的比值来更新个体位置, 提出一种自适应位置更新方程策略. 受粒子群算法启发, Long 等^[16] 引入惯性权重系数、结合个体自身经历过的最优位置和全局最优个体位置, 构建一种修改的位置更新方程策略.

3) 与其他算法的混合. 不同的算法具有不同的搜索能力. 因此, 与其他算法混合也是 GWO 改进的研究方向. 一个较直接的思路是将 GWO 与其他群体智能优化算法混合, 例如: 遗传算法^[23]、粒子群算法^[24]、乌鸦搜索算法^[25]、布谷鸟搜索算法^[26]、地理生物学优化算法^[27] 等. 另一种重要的思路是将 GWO 与其他局部搜索方法混合, 比如: Powell 局部搜索方法^[28]、模式搜索方法^[29].

4) 引入新的算子. Lu 等^[30] 将拓扑结构算子引入到 GWO 中, 提出一种细胞 GWO 算法. 每个个体有自身的拓扑邻居, 个体之间的相互作用仅限于邻居, 有助于增强算法的开采能力. Rodríguez 等^[19] 引入一种新的模糊等级算子, 该算子利用模糊逻辑动态自适应调整每只狼的权重以平衡算法的全局和局部搜索能力. Heidari 等^[31] 引入 Lévy 飞行算子以增强 GWO 算法的全局勘探能力, 避免算法陷入局部最优. Gupta 等^[32] 将随机步行算子引入到 GWO 算法中. 随机步行算子可以增强种群多样性.

虽然大多数改进 GWO 算法引入额外的算子或新的机制以改善性能, 但早熟收敛、全局勘探和局部开采能力不平衡的问题仍然存在. 因此, 本文提出一种新的改进 GWO 算法 (Lens imaging learning grey wolf optimizer, LIL-GWO). 本文的主要贡献描述如下:

- 1) 设计一种修改的控制参数策略以平衡算法的全局勘探和局部开采能力.
- 2) 提出一种基于光学透镜成像原理的反向学习策略以帮助群体跳出局部最优.
- 3) 证明了 LIL-GWO 算法的收敛性.

1 背景知识

1.1 灰狼优化算法

GWO 算法是模仿自然界中灰狼群体的社会等级和捕食行为而衍生出的一种优化方法^[7]. 灰狼群体的社会等级分别为 α 狼、 β 狼、 δ 狼和 ω 狼. 狼群的捕食行为分为三个步骤: 跟踪追捕、包围和攻击

猎物.

狼群包围猎物的数学模型为^[7]:

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \cdot |\mathbf{C} \cdot \mathbf{X}_p(t) - \mathbf{X}(t)| \quad (2)$$

其中, \mathbf{X} 和 \mathbf{X}_p 分别为灰狼个体和猎物的位置向量, t 为当前迭代次数, \mathbf{A} 和 \mathbf{C} 称为系数向量:

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a} \quad (3)$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2 \quad (4)$$

其中, \mathbf{r}_1 和 \mathbf{r}_2 分别为 $[0, 1]$ 间的随机向量, \mathbf{a} 称为距离控制参数, 其值随迭代次数增加而从 2 线性减小到 0, 即

$$\mathbf{a}(t) = 2 - 2 \frac{t}{MaxIter} \quad (5)$$

式中, $MaxIter$ 为最大迭代次数.

狼群攻击猎物的数学模型为^[7]:

$$\mathbf{X}_1 = \mathbf{X}_\alpha - \mathbf{A}_1 \cdot |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}| \quad (6)$$

$$\mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2 \cdot |\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}| \quad (7)$$

$$\mathbf{X}_3 = \mathbf{X}_\delta - \mathbf{A}_3 \cdot |\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}| \quad (8)$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1(t) + \mathbf{X}_2(t) + \mathbf{X}_3(t)}{3} \quad (9)$$

其中, \mathbf{X}_α 、 \mathbf{X}_β 和 \mathbf{X}_δ 分别为 α 、 β 和 δ 狼的位置向量, \mathbf{A}_1 、 \mathbf{A}_2 和 \mathbf{A}_3 的计算类似于 \mathbf{A} 、 \mathbf{C}_1 、 \mathbf{C}_2 和 \mathbf{C}_3 的计算类似于 \mathbf{C} .

基本 GWO 算法的伪代码如算法 1 所示.

算法 1. 基本 GWO 算法.

输入: 种群规模 N , 迭代次数 t , 最大迭代次数 $MaxIter$;

输出: \mathbf{X}_α

1. 在搜索空间中随机初始化 N 个灰狼个体构成初始种群

2. 初始化参数 \mathbf{A} 、 \mathbf{a} 和 \mathbf{C} 的值;

3. 计算每个灰狼个体的适应度值 $f(\mathbf{X}_i)$, $i = 1, 2, \dots, N$;

4. 记录 \mathbf{X}_α 、 \mathbf{X}_β 和 \mathbf{X}_δ 的值, 令 $t = 0$;

5. WHILE $t < MaxIter$;

6. FOR $i = 1$ TO N DO;

7. 根据式 (6)~(9) 更新第 i 只灰狼的位置

8. END FOR

9. 根据式 (4) 计算距离控制参数 \mathbf{a} 的值;

10. 根据式 (2) 和 (3) 更新参数 \mathbf{A} 和 \mathbf{C} 的值;

11. 计算每个个体的适应度值 $f(\mathbf{X}_i)$, $i = 1, 2, \dots, N$;

12. 更新 \mathbf{X}_α 、 \mathbf{X}_β 和 \mathbf{X}_δ 的值;

13. $t = t + 1$;

14. END WHILE

1.2 光的透镜成像原理

凸透镜成像规律是一种光学定律, 它是指将物体放在焦点之外, 在凸透镜另一侧成倒立的实像. 其原理如图 1 所示.

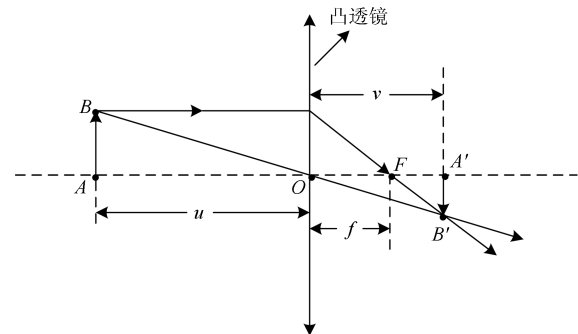


图 1 光的凸透镜成像原理图

Fig. 1 The convex lens image of light

由图 1 可得出透镜成像公式:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \quad (10)$$

其中, u 为物距, v 为像距, f 为透镜焦距.

2 基于透镜成像学习的 GWO 算法

在利用群体智能算法求解复杂高维全局优化问题时, 如何平衡算法的收敛性和种群多样性至关重要. GWO 作为一种群体智能算法, 也需要解决这个问题. 目前, 研究学者为了平衡收敛性和多样性, 提出了一系列的改进 GWO 算法. 没有免费午餐定理 (No free lunch, NFL)^[33] 指出, 任何一种优化算法不可能解决所有类型的优化问题, 只对某些特定的问题有效. 这就是本文改进 GWO 算法的动机所在.

2.1 修改控制参数 \mathbf{C} 策略

由式 (2) 可知, GWO 算法的全局勘探能力主要取决于控制参数 \mathbf{C} 向量^[7,34], \mathbf{C} 的取值是为了能随机增加 ($\mathbf{C} > 1$) 或减轻 ($\mathbf{C} < 1$) 灰狼群体靠近猎物的难易程度. 然而, 目前对控制参数 \mathbf{C} 的研究鲜有文献报道. 为了说明当 $\mathbf{C} > 1$ 和 $\mathbf{C} < 1$ 时, GWO 算法的全局勘探和局部开采能力的转换, 对多个不同类型的测试函数进行实验, 得出相同的结论. 限于篇幅, 图 2 只给出了 GWO 算法求解 2 个标准测试函数 (Sphere 和 Alpine) 的收敛曲线, 其中, Sphere 是单峰函数, Alpine 是多峰函数, 函数维数设置为 $D = 30$, 种群规模为 $N = 30$, 最大迭代次数为 $MaxIter = 500$.

从图 2 可以清晰地看出, 不管是单峰还是多峰函数, 与 ($\mathbf{C} < 1$) 相比, 当 ($\mathbf{C} > 1$) 时, GWO 算法具有较强的局部开采能力和较快的收敛速度. 另外,

在算法搜索前期阶段, ($C > 1$) 也具有较强的全局勘探能力. 而在前期 ($C < 1$) 具有较强的局部开采能力.

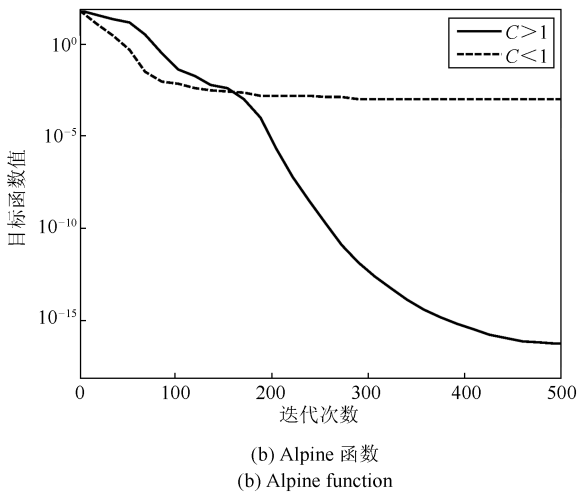
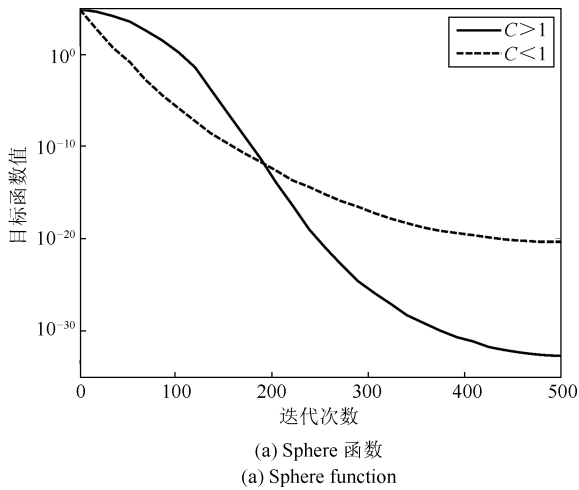


图 2 GWO 算法对两个测试函数的收敛曲线

Fig. 2 Convergence curves of GWO for two test functions

为了平衡 GWO 算法的全局勘探和局部开采能力, 本文提出一种修改的控制参数 C 策略, 其具体表达式如下:

$$C = 2 \times r_3 - a \tag{11}$$

其中, a 的值由式 (5) 计算可得, r_3 是 $[0.5, 1.5]$ 之间的随机向量.

与式 (4) 相比, 式 (11) 具有如下特点: 一般来说, 在算法迭代前期, 种群具有较好的多样性, 此阶段的目的是加快收敛速度, 式 (11) 能较好地满足这一目的. 原因是由于距离控制参数 a 是随迭代次数增加而从 2 线性递减到 0, 因此, 在算法迭代的前期, 控制参数 C 的值小于 1 的几率增大, 这时算法具有较强的开采能力. 随着迭代次数的增加, a 的值逐渐减小, 由式 (11) 可知, 参数 C 的值大于 1 的几率增

大, 加快局部精确搜索显得尤为重要. 因此, 式 (11) 能有效地平衡算法的全局勘探和局部开采能力. 图 3 给出了由式 (4) 和式 (11) 计算得到的参数 C 值的分布, 从图 3 中可以更直观地验证式 (11) 的特点.

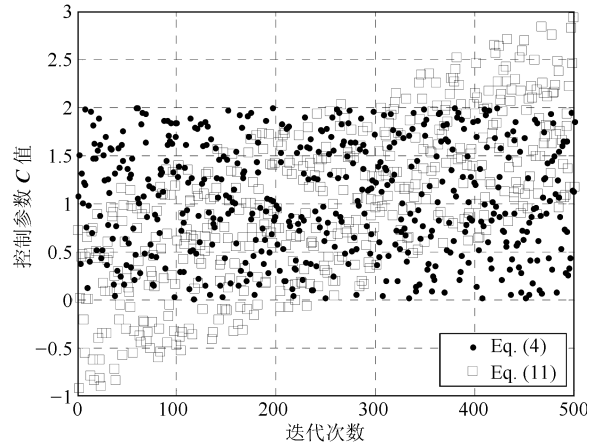


图 3 式 (4) 和式 (11) 计算得到的 C 值

Fig. 3 C values calculated by Eqs. (4) and (11)

2.2 透镜成像反向学习策略

由式 (6)~(9) 可知, 群体中其他狼在 α 狼、 β 狼和 δ 狼的领导下更新自身位置. 然而, 如果 α 、 β 和 δ 均处于局部最优时, 则整个群体均聚集在局部最优区域, 导致种群多样性降低, 算法易陷入局部最优. 这一情形在求解复杂高维优化问题时尤甚. 图 4 给出了 GWO 算法求解 Sphere 函数时 30 个种群在不同迭代次数时的分布情况, 为了更好地显示效果, 问题维数设置为 2 维, 变量的搜索区间设置为 $[-10, 10]$.

从图 4 可以清晰地看出, 在 GWO 算法进化初期, 灰狼个体能较分散地分布在搜索空间中, 这时, 种群多样性较好, 具有较强的全局勘探能力. 随着迭代搜索过程的进行, 群体中其他灰狼个体受 α 、 β 和 δ 狼的引导, 朝最优区域逐渐逼近. 在算法进化搜索末期, 种群中所有灰狼个体均聚集在一个狭小的区域, 导致种群多样性降低, 如果 α 是一个局部最优解, 那么 GWO 算法容易陷入局部最优, 在处理复杂高维优化问题时尤甚. 为了增强 GWO 算法的全局勘探能力, 本文提出一种基于透镜成像原理的反向学习策略, 并将其应用到当前最优个体上产生新的个体, 具体描述过程如下:

定义 1. 反向点^[35]: 假设 $\mathbf{X} = (x_1, x_2, \dots, x_D)$ 为 D 维空间中的一个点, 且 $x_j \in [a_j, b_j], j = 1, 2, \dots, D$, 则 \mathbf{X} 的反向点表示为 $\mathbf{X}' = (x'_1, x'_2, \dots, x'_D)$, 且 $x'_j = a_j + b_j - x_j$.

定义 2. 基点^[36]: 若 D 维空间中存在若干个点 o_1, o_2, \dots, o_m , 对于任意一点 $\mathbf{X} = (x_1, x_2, \dots, x_D)$ 与其反向点 $\mathbf{X}' = (x'_1, x'_2, \dots, x'_D)$ 到 $o_i (i = 1,$

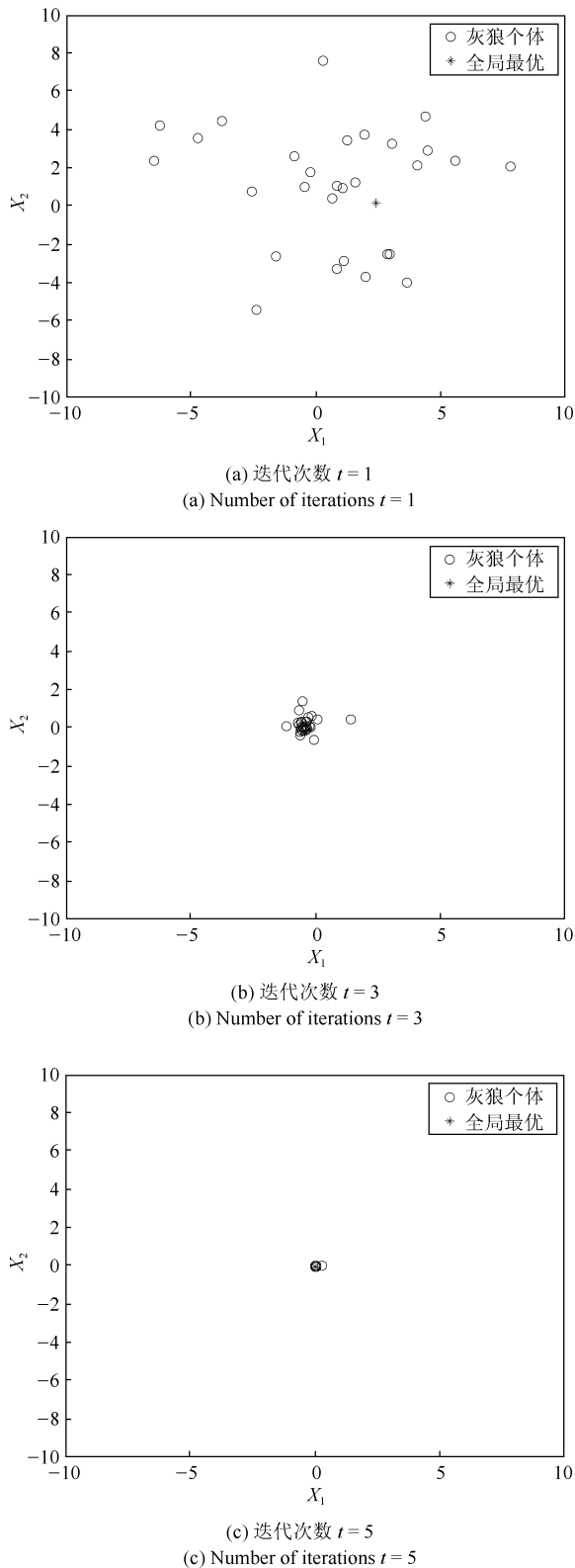


图 4 GWO 算法求解 Sphere 函数 ($D = 2$) 时 30 个种群个体分布情况
Fig. 4 Population distribution observed at various stages in GWO for solving Sphere function ($D = 2$)

$2, \dots, m$) 的欧氏距离分别为 d_i 和 d'_i , 令 $k = d_i/d'_i$, 且 $k = 1, 2, \dots, n$, 则 o_i 被称为 \mathbf{X} 与 \mathbf{X}' 在 $k = i$ 时的基点.

以一维空间为例, 假设有一高度为 h 的个体 P , 它在坐标轴上的投影为 x^* (x^* 为全局最优个体), 基点位置 o (本文取基点为 $[a, b]$ 的中点) 上放置焦距为 f 的透镜, 通过透镜成像过程可得到一个高度为 h' 的像 P' , 它在坐标轴上的投影为 x'^* . 因此, 可得到全局最优个体 x^* 基于透镜成像原理的反向学习策略产生反向个体 x'^* , 如图 5 所示.

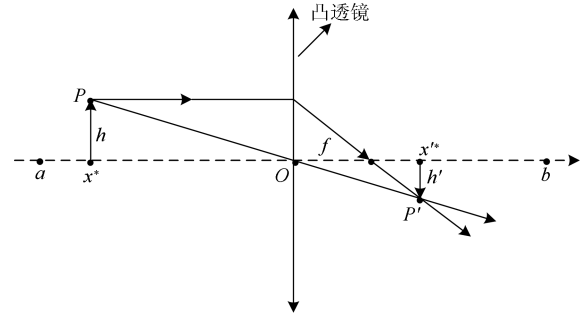


图 5 基于透镜成像的反向学习策略示意图
Fig. 5 Opposition learning strategy based on lens image

在图 5 中, 全局最优个体 x^* 以 o 为基点得到其对应的反向点 x'^* , 由透镜成像原理可以得出:

$$\frac{\frac{a+b}{2} - x^*}{x'^* - \frac{a+b}{2}} = \frac{h}{h'} \quad (12)$$

令 $h/h' = k$, k 称为缩放因子, 对式 (12) 进行变换即可得到反向点的计算公式 x'^* :

$$x'^* = \frac{a+b}{2} + \frac{a+b}{2k} - \frac{x^*}{k} \quad (13)$$

显然, 当 $k = 1$ 时, 式 (13) 可简化为:

$$x'^* = a + b - x^* \quad (14)$$

式 (14) 即为作用在 x^* 上且中心位置在 $[-\frac{a+b}{2}, \frac{a+b}{2}]$ 的一般反向学习策略^[35]. 由式 (13) 和式 (14) 可以清晰地看出, 一般反向学习策略只是透镜成像学习策略的一种特例. 基于一般反向学习策略得到的新候选个体是固定的, 通过调整 k , 基于透镜成像学习策略得到的新候选个体是动态的, 从而更进一步增强群体的多样性.

一般地, 将式 (13) 所示的基于透镜成像原理的反向学习策略推广到 D 维空间可得到:

$$x'_j = \frac{a_j + b_j}{2} + \frac{a_j + b_j}{2k} - \frac{x_j^*}{k} \quad (15)$$

其中, x_j^* 和 x'_j 分别为 x^* 和 x'^* 的第 j 维分量, a_j 和 b_j 分别为决策变量的上下界的第 j 维分量.

2.3 LIL-GWO 算法伪代码

综上所述, LIL-GWO 的伪代码如算法 2 所示.

算法 2. LIL-GWO 算法.

输入: 种群规模 N , 缩放因子 k , 迭代次数 t , 最大迭代次数 $MaxIter$

输出: \mathbf{X}_α

1. 在搜索空间中随机初始化 N 个灰狼个体构成初始种群;
2. 初始化参数 \mathbf{A} 、 \mathbf{a} 和 \mathbf{C} 的值;
3. 计算每个灰狼个体的适应度值 $f(\mathbf{X}_i)$, $i = 1, 2, \dots, N$;
4. 记录 \mathbf{X}_α 、 \mathbf{X}_β 和 \mathbf{X}_δ 的值, 令 $t = 0$;
5. WHILE $t < MaxIter$;
6. FOR $i = 1$ TO N DO;
7. 根据式 (6)~(9) 更新第 i 只灰狼的位置;
8. 利用式 (15) 对 \mathbf{X}_α 进行透镜成像反向学习策略, 产生反向解 \mathbf{X}'_α , 计算 \mathbf{X}'_α 的适应度 $f(\mathbf{X}'_\alpha)$;
9. IF ($f(\mathbf{X}_\alpha) < f(\mathbf{X}'_\alpha)$) DO;
10. 将反向解 \mathbf{X}'_α 替换 \mathbf{X}_α 加入种群参与迭代;
11. END IF
12. END FOR
13. 根据式 (5) 计算距离控制参数 \mathbf{a} 的值;
14. 根据式 (3) 更新参数 \mathbf{A} 的值;
15. 根据式 (11) 更新参数 \mathbf{C} 的值;
16. 计算每个个体的适应度值 $f(\mathbf{X}_i)$, $i = 1, 2, \dots, N$;
17. 更新 \mathbf{X}_α 、 \mathbf{X}_β 和 \mathbf{X}_δ 的值;
18. $t = t + 1$;
19. END WHILE

与基本 GWO 算法相比, LIL-GWO 算法具有以下特点: 1) LIL-GWO 算法没有改变基本 GWO 算法的框架, 仅仅修改了算法参数 \mathbf{C} 和引入新的算子; 2) 与基本 GWO 算法中参数 \mathbf{C} 设置为完全随机数不同, LIL-GWO 算法则引入 \mathbf{a} 改进参数 \mathbf{C} 以平衡全局勘探和局部开采能力; 3) 对当前最优个体执行基于透镜成像的学习策略, 增强种群的多样性, 降低算法陷入局部最优的概率.

2.4 LIL-GWO 算法的收敛性证明

文献 [37] 中给出了基于一般反向学习的群体智能搜索算法收敛性的证明, 这里引入其结论来对 LIL-GWO 算法收敛性进行证明. 需要指出的是, 收敛性证明并不一定能保证算法收敛到全局最优解. 由于 GWO 算法也是群体智能搜索算法, 故有以下定理.

定理 1. 若基于一般反向学习的 GWO 算法收敛, 则 LIL-GWO 算法也是收敛的.

证明. 设 $x_i(t)$ 和 $x'_i(t)$ 分别为第 t 代当前解和

其反向解, $x_{i,j}(t)$ 和 $x'_{i,j}(t)$ 分别为 $x_i(t)$ 和 $x'_i(t)$ 在第 j 维上的值, 问题的全局最优解为 x^* . 由定理 1 中条件可知, 对于第 t 代种群中的解 $x_{i,j}(t)$ 有:

$$\lim_{t \rightarrow \infty} x_{i,j}(t) = x_j^* \quad (16)$$

由于 $a_j(t) = \min(x_{i,j}(t))$, $b_j(t) = \max(x_{i,j}(t))$, 所以得:

$$\lim_{t \rightarrow \infty} a_j(t) = \lim_{b \rightarrow \infty} b_j(t) = x_j^* \quad (17)$$

在第 t 代, 由式 (15) 所示的基于透镜成像原理的反向学习策略产生的当前解的反向解为:

$$x'_{i,j}(t) = \frac{a_j(t) + b_j(t)}{2} + \frac{a_j(t) + b_j(t)}{2k} - \frac{x_{i,j}(t)}{k} \quad (18)$$

当 $t \rightarrow \infty$ 时, 由式 (18) 得:

$$\begin{aligned} \lim_{t \rightarrow \infty} x'_{i,j}(t) &= \lim_{t \rightarrow \infty} \frac{a_j(t) + b_j(t)}{2} + \\ &\lim_{t \rightarrow \infty} \frac{a_j(t) + b_j(t)}{2k} - \lim_{t \rightarrow \infty} \frac{x_{i,j}(t)}{k} = \\ &\frac{x_j^* + x_j^*}{2} + \frac{x_j^* + x_j^*}{2k} - \frac{x_j^*}{k} = x_j^* \end{aligned} \quad (19)$$

由式 (19) 可知, 当 $x_i(t)$ 收敛于 x^* 时, 基于透镜成像原理反向学习策略产生的反向解 $x'_i(t)$ 也收敛于 x^* . 因此, 基于一般反向学习的 GWO 算法收敛, 则 LIL-GWO 算法也收敛. \square

3 实验结果及分析

为了验证 LIL-GWO 算法的有效性, 本文选取两组标准测试函数进行仿真实验, 一组为 12 个国际上通用的标准测试函数, 另一组为来自 IEEE CEC2014 的 30 个标准测试函数.

3.1 标准测试函数及实验环境

从已有文献中选取 12 个国际上通用的标准测试函数, 表 1 给出了这些函数的函数名、具体表达式和搜索区间. 在表 1 中, $f_1 - f_6$ 为单峰函数, 每个函数只有一个全局最优解, 通常用来测试算法的局部开采能力; $f_7 - f_{12}$ 为多峰函数, 主要用来测试算法平衡全局勘探和局部开采的能力. 12 个函数的理论全局最优值均为 0.

在仿真实验中, 所有程序均在 Inter Core Quad, CPU: Q8300、2.50 GHz 主频和 4.00 GB RAM 的计算机上实现, 在 Windows 8 操作系统下, 程序采用 MATLAB 2014a 语言实现.

3.2 与基本 GWO 和改进 GWO 算法的比较

利用 LIL-GWO 算法求解表 1 中的 12 个标准测试函数, 并与基本 GWO 算法、修改 GWO 算法 (Modified GWO, mGWO)^[14]、平均权重 GWO 算法 (Weight average GWO, WAGWO)^[19]、天体物理学启发的 GWO (Astrophysics-inspired GWO, AIGWO) 算法^[34] 和勘探增强的 GWO 算法 (Exploration-enhanced GWO, EEGWO)^[18] 进行比较. 为了比较的公平性, 6 种算法采用相同的适应度函数评价次数 (15 000 次), 即种群规模为 30, 最大迭代次数为 500. GWO、mGWO、WAGWO、AIGWO 和 EEGWO 算法的其他参数设置详见各自原文献. 在 LIL-GWO 算法中, 缩放因子 $k = 10\,000$. 12 个测试函数的维数 $D = 30$, 6 种算法对每个测试函数单独运行 30 次实验, 分别记

录它们的平均值、标准差和基于 Friedman's 排名检验^[18] 的结果, 如表 2 所示.

从表 2 可知, 与 GWO 和 WAGWO 算法相比, 除了函数 f_4 , LIL-GWO 算法分别在其他函数上获得了较好的结果; 然而, GWO 和 WAGWO 算法在函数 f_4 上取得了较好的结果. LIL-GWO 分别在 10 个和 1 个函数上获得了比 mGWO 算法较好和相似的结果; 然而, 对于 f_4 , mGWO 的结果要优. 与 AIGWO 算法相比, LIL-GWO 分别在 9 个和 2 个函数上取得了较好和相似的结果; 与 EEGWO 算法相比, 除了函数 f_4 , LIL-GWO 在其他测试函数上取得了相似的结果; 对于 f_4 , LIL-GWO 得到了较好的结果. 由表 2 中 Friedman's 排名检验结果可知, LIL-GWO 算法性能排名第一, EEGWO 算法排第二, 而基本 GWO 算法性能最差. 限于论文篇幅, 图 6 只给出了 6 种算法对 6 个函数的收敛曲线.

表 1 12 个标准测试函数
Table 1 Twelve benchmark test functions

| 函数名 | 函数表达式 | 搜索空间 |
|------------------|---|-----------------|
| Sphere | $f_1(\mathbf{X}) = \sum_{i=1}^D x_i^2$ | $[-100, 100]$ |
| Schwefel's 2.22 | $f_2(\mathbf{X}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $ | $[-10, 10]$ |
| Schwefel's 2.21 | $f_3(\mathbf{X}) = \max_i \{ x_i , 1 \leq x_i \leq D\}$ | $[-100, 100]$ |
| Rosenbrock | $f_4(\mathbf{X}) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | $[-30, 30]$ |
| Sum-Power | $f_5(\mathbf{X}) = \sum_{i=1}^D x_i ^{(i+1)}$ | $[-1, 1]$ |
| Elliptic | $f_6(\mathbf{X}) = \sum_{i=1}^D (10^6)^{(i-1)/(n-1)} x_i^2$ | $[-100, 100]$ |
| Rastrigin | $f_7(\mathbf{X}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | $[-5.12, 5.12]$ |
| Ackley | $f_8(\mathbf{X}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$ | $[-32, 32]$ |
| Griewank | $f_9(\mathbf{X}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | $[-600, 600]$ |
| Alpine | $f_{10}(\mathbf{X}) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $ | $[-10, 10]$ |
| Levy | $f_{11}(\mathbf{X}) = \sum_{i=1}^D (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_D - 1 [1 + \sin^2(3\pi x_D)]$ | $[-10, 10]$ |
| Stretched V-sine | $f_{12}(\mathbf{X}) = \sum_{i=1}^{D-1} (x_i^2 + 2x_{i+1}^2)^{0.25} \cdot ((\sin 50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1)$ | $[-10, 10]$ |

表 2 LIL-GWO 与其他 5 种算法对 12 个测试函数的结果比较
Table 2 Comparisons of LIL-GWO and other five algorithms for 12 test functions

| 函数 | 统计结果 | GWO | mGWO | WAGWO | AIGWO | EEGWO | LIL-GWO |
|-------|------|------------------------|------------------------|------------------------|------------------------|----------|----------|
| f_1 | 平均值 | 1.36×10^{-29} | 7.89×10^{-44} | 7.66×10^{-35} | 3.62×10^{-42} | 0 | 0 |
| | 标准差 | 1.62×10^{-29} | 8.68×10^{-44} | 1.01×10^{-34} | 3.89×10^{-42} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| f_2 | 平均值 | 4.87×10^{-18} | 1.16×10^{-26} | 6.83×10^{-21} | 1.38×10^{-25} | 0 | 0 |
| | 标准差 | 2.58×10^{-18} | 6.14×10^{-27} | 5.18×10^{-21} | 1.64×10^{-25} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| f_3 | 平均值 | 1.89×10^{-7} | 4.48×10^{-12} | 6.97×10^{-9} | 3.17×10^{-12} | 0 | 0 |
| | 标准差 | 8.81×10^{-8} | 6.38×10^{-12} | 5.29×10^{-9} | 6.34×10^{-12} | 0 | 0 |
| | 排名 | 6 | 4 | 5 | 3 | 1 | 1 |

表 2 LIL-GWO 与其他 5 种算法对 12 个测试函数的结果比较 (续)

Table 2 Comparisons of LIL-GWO and other five algorithms for 12 test functions (Continued)

| 函数 | 统计结果 | GWO | mGWO | WAGWO | AIGWO | EEGWO | LIL-GWO |
|----------|------|-------------------------|--------------------------------------|-------------------------|-------------------------|--|--|
| f_4 | 平均值 | 2.72×10^1 | 2.71×10^1 | 2.75×10^1 | 2.78×10^1 | 2.90×10^1 | 2.89×10^1 |
| | 标准差 | 9.99×10^{-1} | 6.53×10^{-1} | 9.75×10^{-1} | 1.13×10^0 | 5.58×10^{-3} | 7.43×10^{-2} |
| | 排名 | 2 | 1 | 3 | 4 | 6 | 5 |
| f_5 | 平均值 | 1.44×10^{-101} | 4.69×10^{-152} | 4.08×10^{-121} | 4.89×10^{-154} | 0 | 0 |
| | 标准差 | 1.49×10^{-101} | 5.30×10^{-152} | 5.77×10^{-121} | 2.80×10^{-154} | 0 | 0 |
| | 排名 | 6 | 4 | 5 | 3 | 1 | 1 |
| f_6 | 平均值 | 9.01×10^{-26} | 3.12×10^{-40} | 1.14×10^{-31} | 6.74×10^{-40} | 0 | 0 |
| | 标准差 | 2.81×10^{-25} | 3.34×10^{-40} | 9.06×10^{-32} | 1.24×10^{-39} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| f_7 | 平均值 | 2.08×10^0 | 0 | 4.54×10^{-14} | 0 | 0 | 0 |
| | 标准差 | 4.64×10^0 | 0 | 2.54×10^{-14} | 0 | 0 | 0 |
| | 排名 | 6 | 1 | 5 | 1 | 1 | 1 |
| f_8 | 平均值 | 6.84×10^{-14} | 1.23×10^{-14} | 3.29×10^{-14} | 1.37×10^{-14} | 8.88×10^{-16} | 8.88×10^{-16} |
| | 标准差 | 1.08×10^{-14} | 3.89×10^{-15} | 1.97×10^{-15} | 3.18×10^{-15} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| f_9 | 平均值 | 8.52×10^{-13} | 4.44×10^{-16} | 4.17×10^{-15} | 0 | 0 | 0 |
| | 标准差 | 3.03×10^{-13} | 0 | 1.69×10^{-15} | 0 | 0 | 0 |
| | 排名 | 6 | 4 | 5 | 1 | 1 | 1 |
| f_{10} | 平均值 | 5.25×10^{-4} | 1.09×10^{-24} | 1.05×10^{-4} | 9.70×10^{-22} | 0 | 0 |
| | 标准差 | 5.34×10^{-4} | 2.18×10^{-24} | 2.35×10^{-4} | 2.13×10^{-21} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| f_{11} | 平均值 | 2.59×10^{-31} | 4.64×10^{-45} | 6.40×10^{-36} | 2.32×10^{-44} | 0 | 0 |
| | 标准差 | 1.34×10^{-31} | 7.73×10^{-45} | 8.76×10^{-36} | 5.97×10^{-44} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| f_{12} | 平均值 | 6.12×10^{-8} | 3.21×10^{-12} | 3.82×10^{-9} | 1.75×10^{-11} | 0 | 0 |
| | 标准差 | 1.74×10^{-8} | 2.24×10^{-12} | 1.37×10^{-9} | 1.33×10^{-11} | 0 | 0 |
| | 排名 | 6 | 3 | 5 | 4 | 1 | 1 |
| | 平均排名 | 5.6667 | 2.9167 | 4.8333 | 3.3333 | 1.4167 | 1.3333 |
| | 最终排名 | 6 | 3 | 5 | 4 | 2 | 1 |

从图 6 可以清晰地看出, 与 GWO、mGWO、WAGWO 和 AIGWO 算法相比, LIL-GWO 算法在 6 个测试函数上具有较高的求解精度和较快的收敛速度. 虽然 EEGWO 算法在 6 个测试函数上获得了与 LIL-GWO 算法相似的精度, 但其收敛速度明显要慢于 LIL-GWO 算法; 对于 f_8 , 两种算法取得了相似的求解精度和收敛速度.

3.3 与其他群体智能优化算法的比较

为了进一步验证其有效性, 将 LIL-GWO 与 8 种其他群体智能优化算法进行比较, 它们是协方差矩阵自适应进化策略 (Covariance matrix adaptation, CMA-ES)^[38]、改进粒子群优化 (Improved particle swarm optimization, IPSO) 算法^[39]、基于

反向学习的差分进化 (Opposition-based differential evolution, ODE) 算法^[35]、全局最优指导人工蜂群 (Gbest-guided artificial bee colony, GABC) 算法^[40]、精英教与学优化算法 (Elitist teaching-learning-based optimization, ETLBO)^[41]、改进鲸鱼优化算法 (Improved whale optimization algorithm, IWOA)^[42]、引入惯性权重的改进正弦余弦算法 (Improved sine cosine algorithm, ISCA)^[43]. 为了比较的公平性, 8 种算法采用相同的适应度评价次数 15000 次, 即种群规模为 30, 最大迭代次数为 500. 函数的维数设置为 $D = 30$. 表 3 和表 4 分别给出了 8 种算法对 12 个测试函数 30 次独立实验获得的平均值、标准差、Friedman's 排名检验结果、多问题 Wilcoxon's 非参数检验结果和 Wilcoxon's

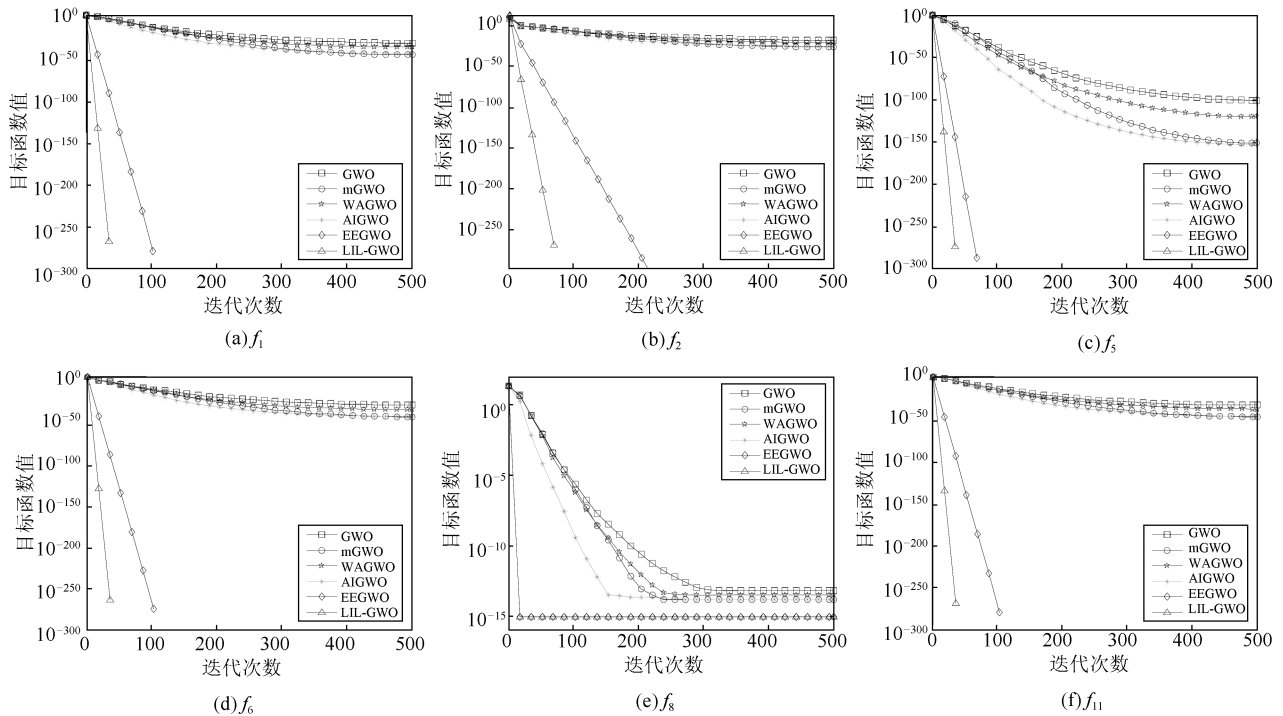


图6 6种算法对6个代表性测试函数的收敛曲线

Fig. 6 Convergence curves of six algorithms for six representative test functions

秩和检验结果.

从表3中比较结果可知,与CMA-ES、ODE和GABC算法相比,LIL-GWO在11个测试函数上获得了较好的结果.然而,对于函数 f_4 ,CMA-ES、ODE和GABC取得了较好的结果.与IPSO相比,LIL-GWO在所有测试函数上得到了较好的性能.与ETLBO相比,LIL-GWO在10个测试函数上取得了较好的性能和1个函数(f_9)上获得了相似的性能;然而,对于 f_4 ,ETLBO得到了稍优的结果.与IWOA相比,LIL-GWO在3个测试函数上获得了较好的结果和8个函数上取得了相似的结果;对于 f_4 ,IWOA的结果要优.ISCA是最近提出的一种具有较强竞争力群体智能算法,与ISCA相比,LIL-GWO在11个测试函数上取得了相似的性能;对于 f_4 ,ISCA得到稍差的结果.从Friedman's排名检验结果可以看出,LIL-GWO排名第一,ISCA排名第二,依次为IWOA、ETLBO、ODE、GABC、CMA-ES和IPSO.

由表4可以看出,LIL-GWO算法在所有其他对比情况下所获得的 R^+ 值均大于 R^- 值.根据Wilcoxon's秩和检验,除了LIL-GWO与IWOA相比、LIL-GWO与ISCA相比两种情形,其他对比算法的 p 值均小于0.05,这说明LIL-GWO算法的性能明显优于CMA-ES、IPSO、ODE、GABC和ETLBO算法.另外,LIL-GWO与ISCA和IWOA算法的性能没有明显的差异性.总体来说,与其他群

体智能算法相比,LIL-GWO算法的性能具有较强的竞争力.

3.4 与一般反向学习策略的比较

为了比较基于透镜成像学习(Lens image based learning, LIBL)和一般反向学习(Oppositional based learning, OBL)策略的有效性,将它们嵌入到基本GWO算法中,分别得到LIBL-GWO和OBL-GWO算法.本小节选取表1中的12个测试函数进行仿真实验来比较两种算法性能的优劣.参数设置与第3.2节相同.表5给出了OBL-GWO和LIBL-GWO算法对12个测试函数($D=30$ 维)30次独立实验获得的平均值和标准差结果.

从表5中结果可知,与OBL-GWO算法相比,除了 f_4 、 f_7 和 f_9 ,LIBL-GWO算法在其他9个测试函数上均获得了明显的优势;对于 f_7 和 f_9 ,两种算法取得了相似的实验结果;然而,OBL-GWO算法在函数 f_4 上得到了稍优的结果.这表明GWO算法采用基于透镜成像学习的策略比采用一般反向学习策略更能避免算法陷入局部最优.

3.5 LIL-GWO算法求解CEC 2014测试问题

为了更进一步验证算法的有效性和可行性,将LIL-GWO算法应用于求解IEEE CEC2014测试集.CEC2014测试集包含30个函数,它们比表1中的测试函数更复杂、求解更困难.30个测试函数

表 3 LIL-GWOLIL-GWO 与其他 7 种算法对 12 个函数的结果比较

Table 3 Comparisons of LIL-GWOLIL-GWO and other seven algorithms for 12 test functions

| 函数 | 统计结果 | CMA-ES | IPSO | ODE | GABC | ETLBO | IWOA | ISCA | LIL-GWO |
|----------|------|--------------------------------------|------------------------|-------------------------|------------------------|-------------------------|--|--|--|
| f_1 | 平均值 | 8.64×10^{-11} | 2.82×10^{-16} | 2.68×10^{-49} | 4.00×10^{-16} | 2.70×10^{-119} | 0 | 0 | 0 |
| | 标准差 | 3.83×10^{-11} | 5.07×10^{-16} | 2.50×10^{-49} | 3.76×10^{-16} | 4.29×10^{-119} | 0 | 0 | 0 |
| | 排名 | 8 | 6 | 5 | 7 | 4 | 1 | 1 | 1 |
| f_2 | 平均值 | 2.03×10^{-5} | 4.03×10^{-3} | 3.86×10^{-31} | 2.59×10^{-7} | 1.19×10^{-60} | 2.77×10^{-267} | 0 | 0 |
| | 标准差 | 1.15×10^{-5} | 8.05×10^{-3} | 4.00×10^{-31} | 1.98×10^{-8} | 5.87×10^{-61} | 0 | 0 | 0 |
| | 排名 | 7 | 8 | 5 | 6 | 4 | 3 | 1 | 1 |
| f_3 | 平均值 | 1.38×10^{-4} | 2.13×10^0 | 1.47×10^{-2} | 1.16×10^{-1} | 2.35×10^{-36} | 6.35×10^{-95} | 0 | 0 |
| | 标准差 | 2.59×10^{-5} | 7.32×10^{-1} | 2.66×10^{-3} | 2.27×10^{-2} | 1.82×10^{-36} | 1.26×10^{-94} | 0 | 0 |
| | 排名 | 5 | 8 | 6 | 7 | 4 | 3 | 1 | 1 |
| f_4 | 平均值 | 1.83×10^1 | 7.98×10^1 | 2.81×10^1 | 2.86×10^1 | 2.50×10^1 | 2.87×10^1 | 2.90×10^1 | 2.89×10^1 |
| | 标准差 | 3.56×10^{-1} | 5.57×10^1 | 3.45×10^{-1} | 1.66×10^{-1} | 2.65×10^{-1} | 5.87×10^{-2} | 4.55×10^{-1} | 7.43×10^{-2} |
| | 排名 | 1 | 8 | 3 | 4 | 2 | 5 | 7 | 6 |
| f_5 | 平均值 | 3.91×10^{-10} | 2.17×10^{-34} | 8.51×10^{-149} | 4.02×10^{-42} | 3.26×10^{-278} | 0 | 0 | 0 |
| | 标准差 | 4.00×10^{-10} | 4.21×10^{-34} | 8.47×10^{-149} | 6.81×10^{-42} | 0 | 0 | 0 | 0 |
| | 排名 | 8 | 7 | 5 | 6 | 4 | 1 | 1 | 1 |
| f_6 | 平均值 | 2.89×10^{-3} | 5.28×10^{-8} | 1.91×10^{-44} | 1.86×10^{-12} | 1.18×10^{-115} | 0 | 0 | 0 |
| | 标准差 | 2.58×10^{-3} | 6.99×10^{-8} | 2.23×10^{-44} | 1.02×10^{-12} | 1.30×10^{-115} | 0 | 0 | 0 |
| | 排名 | 8 | 7 | 5 | 6 | 4 | 1 | 1 | 1 |
| f_7 | 平均值 | 1.26×10^2 | 2.57×10^1 | 1.14×10^{-14} | 9.62×10^{-15} | 8.96×10^0 | 0 | 0 | 0 |
| | 标准差 | 6.85×10^1 | 1.54×10^0 | 2.54×10^{-14} | 1.88×10^{-14} | 6.17×10^0 | 0 | 0 | 0 |
| | 排名 | 8 | 7 | 5 | 4 | 6 | 1 | 1 | 1 |
| f_8 | 平均值 | 2.41×10^{-6} | 7.10×10^{-7} | 1.07×10^{-14} | 3.81×10^{-14} | 2.66×10^{-15} | 8.88×10^{-16} | 8.88×10^{-16} | 8.88×10^{-16} |
| | 标准差 | 6.98×10^{-7} | 1.54×10^{-6} | 1.85×10^{-15} | 2.16×10^{-15} | 9.93×10^{-16} | 0 | 0 | 0 |
| | 排名 | 8 | 7 | 5 | 6 | 4 | 1 | 1 | 1 |
| f_9 | 平均值 | 6.60×10^{-11} | 2.33×10^{-1} | 2.44×10^{-16} | 6.21×10^{-16} | 0 | 0 | 0 | 0 |
| | 标准差 | 1.32×10^{-11} | 1.30×10^{-1} | 1.45×10^{-16} | 9.63×10^{-16} | 0 | 0 | 0 | 0 |
| | 排名 | 7 | 8 | 5 | 6 | 1 | 1 | 1 | 1 |
| f_{10} | 平均值 | 9.71×10^{-6} | 6.32×10^{-5} | 9.53×10^0 | 1.35×10^{-6} | 2.21×10^{-61} | 7.28×10^{-262} | 0 | 0 |
| | 标准差 | 2.94×10^{-6} | 7.25×10^{-5} | 7.79×10^0 | 2.42×10^{-6} | 2.97×10^{-61} | 0 | 0 | 0 |
| | 排名 | 6 | 7 | 8 | 5 | 4 | 3 | 1 | 1 |
| f_{11} | 平均值 | 4.59×10^{-5} | 1.19×10^1 | 2.38×10^{-49} | 4.58×10^{-15} | 2.62×10^{-120} | 0 | 0 | 0 |
| | 标准差 | 4.27×10^{-5} | 8.43×10^0 | 3.10×10^{-49} | 6.63×10^{-15} | 3.04×10^{-120} | 0 | 0 | 0 |
| | 排名 | 7 | 8 | 5 | 6 | 4 | 1 | 1 | 1 |
| f_{12} | 平均值 | 1.49×10^{-1} | 1.53×10^1 | 3.51×10^{-15} | 8.06×10^{-2} | 2.08×10^{-28} | 0 | 0 | 0 |
| | 标准差 | 5.87×10^{-3} | 6.60×10^0 | 2.45×10^{-15} | 1.85×10^{-2} | 1.35×10^{-28} | 0 | 0 | 0 |
| | 排名 | 7 | 8 | 5 | 6 | 4 | 1 | 1 | 1 |
| 平均排名 | | 6.6667 | 7.4167 | 5.1667 | 5.7500 | 3.7500 | 1.9167 | 1.5000 | 1.4167 |
| 最终排名 | | 7 | 8 | 5 | 6 | 4 | 3 | 2 | 1 |

表 4 LIL-GWO 与其他 7 种算法的统计检验结果比较

Table 4 Statistical test results of LIL-GWO and other seven algorithms

| 算法 | R+ | R- | p-value | $\alpha = 0.05$ | $\alpha = 0.1$ |
|-----------------------|------|------|-------------------------|-----------------|----------------|
| LIL-GWO versus CMA-ES | 67.0 | 11.0 | 2.8848×10^{-4} | Yes | Yes |
| LIL-GWO versus IPSO | 78.0 | 0.0 | 4.5561×10^{-4} | Yes | Yes |
| LIL-GWO versus ODE | 67.0 | 11.0 | 1.3461×10^{-3} | Yes | Yes |
| LIL-GWO versus GABC | 66.0 | 12.0 | 7.0988×10^{-4} | Yes | Yes |
| LIL-GWO versus ETLBO | 62.0 | 16.0 | 5.7382×10^{-3} | Yes | Yes |
| LIL-GWO versus IWOA | 40.0 | 38.0 | 3.1412×10^{-1} | No | No |
| LIL-GWO versus ISCA | 39.5 | 38.5 | 9.8936×10^{-1} | No | No |

表 5 两种算法对 12 个函数的实验结果比较
Table 5 Experimental results of two algorithms for
12 functions

| 函数 | OBL-GWO | | LIBL-GWO | |
|----------|--------------------------------------|-------------------------|--|-----------------------|
| | 平均值 | 标准差 | 平均值 | 标准差 |
| f_1 | 2.37×10^{-35} | 3.66×10^{-35} | 0 | 0 |
| f_2 | 1.27×10^{-18} | 7.90×10^{-19} | 0 | 0 |
| f_3 | 2.50×10^{-33} | 4.36×10^{-33} | 0 | 0 |
| f_4 | 2.87×10^1 | 1.55×10^{-1} | 2.89×10^1 | 5.83×10^{-2} |
| f_5 | 1.16×10^{-145} | 1.60×10^{-145} | 0 | 0 |
| f_6 | 1.46×10^{-28} | 2.05×10^{-28} | 0 | 0 |
| f_7 | 0 | 0 | 0 | 0 |
| f_8 | 4.44×10^{-15} | 0 | 8.88×10^{-16} | 0 |
| f_9 | 0 | 0 | 0 | 0 |
| f_{10} | 5.91×10^{-19} | 8.32×10^{-19} | 0 | 0 |
| f_{11} | 2.90×10^{-34} | 5.73×10^{-34} | 0 | 0 |
| f_{12} | 1.68×10^{-8} | 1.46×10^{-8} | 0 | 0 |

分为 4 类: 单峰函数 (F1~F3)、多峰函数 (F4~F16)、混合函数 (F17~F22) 和复杂的合成函数 (F23~F30). 这些函数的具体表达式详见文献 [44]. 每个测试函数的维数设置为 $D = 30$, 每个决策变量的取值范围为 $[-100, 100]$.

利用 LIL-GWO 算法对 CEC2014 测试集中的 30 个函数进行求解, 并与 7 种较先进的群体智能优化技术进行比较, 它们是: 和声搜索和模拟退火混合算法 (Harmony search and simulated annealing, HS-SA)^[45]、甲壳虫 (Pity beetle algorithm, PBA) 算法^[46]、组合型差分进化 (Composite DE, CoDE) 算法^[47]、修改的人工蜂群 (Modified ABC, MoABC) 算法^[48]、基于动态组策略的教与学优化算法 (Dynamic group strategy TLBO, DGS-TLBO)^[49]、混合布谷鸟搜索算法 (Hybrid cuckoo search algorithm, HCSA)^[50] 和基于拉普拉斯迁移算子的生物地理学优化算法 (Laplacian migration biogeography optimization, LM-BBO)^[51]. 7 种群体智能优化技术代表了 HS、PBA、DE、ABC、TLBO、CSA 和 BBO, 且它们的性能具有较强的竞争力. 7 种算法的参数设置与它们各自的原文献相同. 为了进行公平的比较, LIL-GWO 与其他 7 种算法的最大适应度函数评价次数设置相同, 即为 300 000 次. 每种算法对每个测试函数单独运行 30 次实验, 并分别记录它们的平均误差值、标准差和 Fridman's 统计排名检验结果, 如表 6 所示.

表 6 8 种算法对 CEC 2014 测试集 30 个函数的实验结果比较
Table 6 Comparisons of eight algorithms for 30 test functions from CEC 2014

| 函数 | 统计结果 | HS-SA | PBA | CoDE | MoABC | DGS-TLBO | HSCA | LM-BBO | LIL-GWO |
|----|------|---|-----------------------|-----------------------|-----------------------|--------------------------------------|-----------------------|--------------------------------------|--------------------------------------|
| F1 | 平均误差 | 1.16×10^7 | 3.50×10^7 | 1.21×10^7 | 2.81×10^7 | 1.04×10^7 | 3.50×10^7 | 1.01×10^7 | 2.59×10^7 |
| | 标准差 | 7.89×10^4 | 2.16×10^7 | 4.48×10^6 | 1.01×10^7 | 8.61×10^6 | 2.49×10^7 | 3.81×10^6 | 4.30×10^6 |
| | 排名 | 3 | 7 | 4 | 6 | 2 | 7 | 1 | 3 |
| F2 | 平均误差 | 1.38×10^4 | 3.05×10^8 | 1.89×10^7 | 2.88×10^4 | 4.59×10^6 | 1.95×10^7 | 5.34×10^4 | 1.02×10^7 |
| | 标准差 | 1.34×10^4 | 1.89×10^8 | 9.45×10^6 | 4.11×10^4 | 1.11×10^7 | 5.49×10^7 | 2.14×10^4 | 2.71×10^6 |
| | 排名 | 1 | 8 | 6 | 2 | 4 | 7 | 3 | 5 |
| F3 | 平均误差 | 6.31×10^3 | 6.97×10^3 | 4.16×10^3 | 1.06×10^4 | 1.44×10^1 | 3.10×10^4 | 1.64×10^4 | 2.24×10^4 |
| | 标准差 | 6.06×10^3 | 3.96×10^3 | 1.89×10^3 | 3.66×10^3 | 1.68×10^1 | 1.36×10^4 | 1.71×10^4 | 2.72×10^4 |
| | 排名 | 3 | 4 | 2 | 5 | 1 | 8 | 6 | 7 |
| F4 | 平均误差 | 1.11×10^2 | 5.78×10^2 | 1.44×10^2 | 1.59×10^2 | 1.46×10^2 | 2.03×10^2 | 9.99×10^1 | 2.38×10^2 |
| | 标准差 | 4.01×10^1 | 3.62×10^1 | 1.55×10^1 | 2.76×10^1 | 3.78×10^1 | 6.69×10^1 | 2.85×10^1 | 4.57×10^1 |
| | 排名 | 2 | 8 | 3 | 5 | 4 | 6 | 1 | 7 |
| F5 | 平均误差 | 2.00×10^1 | 5.21×10^2 | 2.10×10^1 | 2.04×10^1 | 2.10×10^1 | 2.00×10^1 | 3.06×10^0 | 2.04×10^1 |
| | 标准差 | 3.01×10^{-4} | 5.26×10^{-2} | 6.56×10^{-2} | 3.53×10^{-2} | 4.34×10^{-2} | 2.28×10^{-3} | 7.87×10^{-1} | 6.42×10^{-2} |
| | 排名 | 2 | 8 | 6 | 4 | 6 | 2 | 1 | 4 |
| F6 | 平均误差 | 1.31×10^1 | 6.16×10^2 | 5.57×10^1 | 3.78×10^1 | 1.67×10^1 | 3.23×10^1 | 1.69×10^1 | 1.25×10^1 |
| | 标准差 | 2.12×10^0 | 2.40×10^0 | 2.67×10^0 | 2.65×10^0 | 3.45×10^0 | 3.27×10^0 | 3.12×10^0 | 2.10×10^0 |
| | 排名 | 2 | 8 | 7 | 6 | 3 | 5 | 4 | 1 |
| F7 | 平均误差 | 1.52×10^{-2} | 7.04×10^2 | 1.20×10^0 | 5.72×10^{-1} | 1.01×10^0 | 1.79×10^0 | 1.76×10^{-1} | 7.66×10^0 |
| | 标准差 | 1.63×10^{-2} | 1.72×10^0 | 7.20×10^{-2} | 1.36×10^{-1} | 1.50×10^0 | 2.19×10^0 | 8.56×10^{-2} | 4.50×10^0 |
| | 排名 | 1 | 8 | 5 | 3 | 4 | 6 | 2 | 7 |

表 6 8 种算法对 CEC 2014 测试集 30 个函数的实验结果比较 (续 1)

Table 6 Comparisons of eight algorithms for 30 test functions from CEC 2014 (Continued 1)

| 函数 | 统计结果 | HS-SA | PBA | CoDE | MoABC | DGS-TLBO | HSCA | LM-BBO | LIL-GWO |
|-----|------|---|-----------------------|--------------------------------------|-----------------------|---|-----------------------|---|---|
| F8 | 平均误差 | 4.10×10^{-5} | 8.56×10^2 | 2.30×10^2 | 1.26×10^1 | 7.67×10^1 | 1.71×10^2 | 5.53×10^1 | 3.49×10^1 |
| | 标准差 | 8.13×10^{-6} | 1.48×10^1 | 1.45×10^1 | 1.74×10^0 | 2.45×10^1 | 3.46×10^1 | 3.78×10^2 | 1.20×10^1 |
| | 排名 | 1 | 8 | 7 | 2 | 5 | 6 | 4 | 3 |
| F9 | 平均误差 | 6.71×10^1 | 1.01×10^3 | 3.80×10^2 | 2.58×10^2 | 9.84×10^1 | 2.80×10^2 | 7.66×10^1 | 6.02×10^1 |
| | 标准差 | 1.52×10^1 | 1.33×10^1 | 1.89×10^1 | 2.83×10^1 | 3.08×10^1 | 5.16×10^1 | 1.61×10^1 | 8.59×10^0 |
| | 排名 | 2 | 8 | 7 | 5 | 4 | 6 | 3 | 1 |
| F10 | 平均误差 | 2.01×10^{-1} | 1.89×10^3 | 7.26×10^3 | 2.29×10^2 | 2.39×10^3 | 2.66×10^3 | 1.26×10^4 | 1.72×10^3 |
| | 标准差 | 4.66×10^{-2} | 3.68×10^2 | 3.84×10^2 | 1.07×10^2 | 4.71×10^2 | 5.34×10^2 | 1.16×10^4 | 1.81×10^2 |
| | 排名 | 1 | 4 | 7 | 2 | 5 | 6 | 8 | 3 |
| F11 | 平均误差 | 1.99×10^3 | 4.49×10^3 | 1.21×10^4 | 5.74×10^3 | 3.93×10^3 | 4.13×10^3 | 1.23×10^4 | 2.64×10^3 |
| | 标准差 | 4.34×10^2 | 5.35×10^2 | 4.27×10^2 | 3.27×10^2 | 5.45×10^2 | 5.35×10^2 | 3.42×10^2 | 3.12×10^2 |
| | 排名 | 1 | 5 | 7 | 6 | 3 | 4 | 8 | 2 |
| F12 | 平均误差 | 2.46×10^{-2} | 1.20×10^3 | 2.47×10^0 | 4.71×10^{-1} | 2.75×10^0 | 5.11×10^{-1} | 1.11×10^{-2} | 3.20×10^{-1} |
| | 标准差 | 1.26×10^{-2} | 1.43×10^{-1} | 2.74×10^{-1} | 5.73×10^{-2} | 2.62×10^{-1} | 2.56×10^{-1} | 1.75×10^{-18} | 3.19×10^{-1} |
| | 排名 | 2 | 8 | 6 | 5 | 7 | 4 | 1 | 3 |
| F13 | 平均误差 | 5.24×10^{-1} | 1.30×10^3 | 6.53×10^{-1} | 4.51×10^{-1} | 4.71×10^{-1} | 4.81×10^{-1} | 6.55×10^{-1} | 3.40×10^{-1} |
| | 标准差 | 1.04×10^{-1} | 9.49×10^{-2} | 6.56×10^{-2} | 4.11×10^{-2} | 1.31×10^{-1} | 1.17×10^{-1} | 1.56×10^{-1} | 5.48×10^{-2} |
| | 排名 | 5 | 8 | 6 | 2 | 3 | 4 | 7 | 1 |
| F14 | 平均误差 | 4.15×10^{-1} | 1.40×10^3 | 4.31×10^{-1} | 2.98×10^{-1} | 2.88×10^{-1} | 3.08×10^{-1} | 6.20×10^{-1} | 4.10×10^{-1} |
| | 标准差 | 2.29×10^{-1} | 4.57×10^{-2} | 8.50×10^{-2} | 2.50×10^{-2} | 4.92×10^{-2} | 5.64×10^{-2} | 2.96×10^{-1} | 2.68×10^{-2} |
| | 排名 | 5 | 8 | 6 | 2 | 1 | 3 | 7 | 4 |
| F15 | 平均误差 | 1.64×10^1 | 1.52×10^3 | 3.78×10^1 | 3.14×10^1 | 3.75×10^1 | 9.80×10^1 | 1.55×10^1 | 1.68×10^0 |
| | 标准差 | 1.17×10^1 | 3.36×10^0 | 2.26×10^0 | 6.02×10^0 | 2.19×10^1 | 3.02×10^1 | 5.50×10^0 | 4.92×10^{-1} |
| | 排名 | 3 | 8 | 6 | 4 | 5 | 7 | 2 | 1 |
| F16 | 平均误差 | 1.42×10^1 | 1.63×10^3 | 2.28×10^1 | 1.97×10^1 | 1.11×10^1 | 1.27×10^1 | 1.08×10^1 | 1.03×10^1 |
| | 标准差 | 7.83×10^{-1} | 3.78×10^{-1} | 3.26×10^{-1} | 4.02×10^{-1} | 6.62×10^{-1} | 5.01×10^{-1} | 5.84×10^{-1} | 9.04×10^{-1} |
| | 排名 | 5 | 8 | 7 | 6 | 3 | 4 | 2 | 1 |
| F17 | 平均误差 | 2.09×10^6 | 3.40×10^6 | 1.81×10^5 | 1.01×10^7 | 1.67×10^5 | 1.48×10^6 | 1.46×10^6 | 1.29×10^6 |
| | 标准差 | 1.31×10^6 | 2.12×10^6 | 1.24×10^5 | 4.96×10^6 | 2.13×10^5 | 1.21×10^6 | 9.34×10^5 | 1.23×10^6 |
| | 排名 | 6 | 7 | 2 | 8 | 1 | 5 | 4 | 3 |
| F18 | 平均误差 | 6.16×10^3 | 1.70×10^6 | 3.62×10^3 | 9.92×10^3 | 8.71×10^2 | 7.67×10^3 | 2.90×10^3 | 4.07×10^5 |
| | 标准差 | 6.22×10^3 | 1.06×10^6 | 2.31×10^3 | 9.94×10^3 | 1.02×10^3 | 6.70×10^3 | 4.27×10^3 | 7.27×10^5 |
| | 排名 | 4 | 8 | 3 | 6 | 1 | 5 | 2 | 7 |
| F19 | 平均误差 | 1.89×10^1 | 1.91×10^3 | 3.62×10^1 | 3.33×10^1 | 2.71×10^1 | 5.33×10^1 | 5.19×10^3 | 1.84×10^1 |
| | 标准差 | 2.46×10^1 | 5.23×10^0 | 1.08×10^1 | 1.06×10^1 | 2.86×10^1 | 3.63×10^1 | 5.67×10^3 | 3.78×10^0 |
| | 排名 | 2 | 7 | 5 | 4 | 3 | 6 | 8 | 1 |
| F20 | 平均误差 | 6.77×10^3 | 8.77×10^3 | 5.04×10^2 | 3.96×10^4 | 4.28×10^2 | 3.93×10^4 | 2.61×10^4 | 1.76×10^4 |
| | 标准差 | 5.09×10^3 | 4.31×10^3 | 3.17×10^2 | 1.29×10^4 | 1.77×10^2 | 2.20×10^4 | 1.56×10^4 | 9.12×10^3 |
| | 排名 | 3 | 4 | 2 | 8 | 1 | 7 | 6 | 5 |
| F21 | 平均误差 | 5.27×10^5 | 4.39×10^5 | 2.12×10^4 | 7.30×10^6 | 2.20×10^4 | 3.54×10^5 | 1.11×10^6 | 3.27×10^5 |
| | 标准差 | 3.59×10^5 | 3.40×10^5 | 1.61×10^4 | 4.36×10^6 | 2.22×10^4 | 3.48×10^5 | 7.95×10^5 | 2.86×10^5 |
| | 排名 | 6 | 5 | 1 | 8 | 2 | 4 | 7 | 3 |
| F22 | 平均误差 | 4.88×10^2 | 2.53×10^3 | 1.44×10^3 | 1.14×10^3 | 3.14×10^2 | 9.47×10^2 | 1.88×10^3 | 1.69×10^3 |
| | 标准差 | 9.15×10^1 | 1.08×10^2 | 1.59×10^2 | 1.89×10^2 | 1.41×10^2 | 3.31×10^2 | 2.04×10^2 | 2.13×10^2 |
| | 排名 | 2 | 8 | 5 | 4 | 1 | 3 | 7 | 6 |

表 6 8 种算法对 CEC 2014 测试集 30 个函数的实验结果比较 (续 2)

Table 6 Comparisons of eight algorithms for 30 test functions from CEC 2014 (Continued 2)

| 函数 | 统计结果 | HS-SA | PBA | CoDE | MoABC | DGS-TLBO | HSCA | LM-BBO | LIL-GWO |
|-----|------|-----------------------|--------------------|-----------------------|-----------------------|--------------------------------------|-----------------------|--------------------------------------|--------------------------------------|
| F23 | 平均误差 | 3.16×10^2 | 2.60×10^3 | 3.55×10^2 | 3.57×10^2 | 3.15×10^2 | 3.29×10^2 | 4.11×10^2 | 2.00×10^2 |
| | 标准差 | 5.74×10^{-1} | 3.88×10^1 | 1.77×10^{-1} | 7.30×10^0 | 4.43×10^{-1} | 7.51×10^0 | 6.43×10^1 | 0 |
| | 排名 | 3 | 8 | 5 | 6 | 2 | 4 | 7 | 1 |
| F24 | 平均误差 | 2.31×10^2 | 2.61×10^3 | 2.83×10^2 | 2.71×10^2 | 2.00×10^2 | 2.78×10^2 | 1.48×10^4 | 2.00×10^2 |
| | 标准差 | 5.17×10^0 | 3.98×10^0 | 1.80×10^0 | 1.78×10^0 | 9.68×10^{-4} | 3.11×10^1 | 8.37×10^3 | 0 |
| | 排名 | 3 | 7 | 6 | 4 | 1 | 5 | 8 | 1 |
| F25 | 平均误差 | 2.05×10^2 | 2.70×10^3 | 2.18×10^2 | 2.22×10^2 | 2.02×10^2 | 2.23×10^2 | 5.29×10^2 | 2.00×10^2 |
| | 标准差 | 1.51×10^0 | 1.41×10^0 | 1.94×10^0 | 2.80×10^0 | 3.62×10^2 | 9.39×10^0 | 4.37×10^1 | 0 |
| | 排名 | 3 | 8 | 6 | 4 | 5 | 7 | 2 | 1 |
| F26 | 平均误差 | 1.38×10^2 | 2.70×10^3 | 1.04×10^2 | 1.01×10^2 | 1.10×10^2 | 1.00×10^2 | 2.13×10^0 | 1.00×10^2 |
| | 标准差 | 4.84×10^1 | 1.93×10^1 | 1.82×10^1 | 6.75×10^{-2} | 3.15×10^1 | 1.63×10^{-1} | 3.46×10^0 | 0 |
| | 排名 | 5 | 8 | 7 | 6 | 3 | 4 | 2 | 1 |
| F27 | 平均误差 | 6.69×10^2 | 3.14×10^3 | 1.28×10^3 | 1.08×10^3 | 7.94×10^2 | 4.27×10^2 | 1.96×10^2 | 2.00×10^2 |
| | 标准差 | 1.63×10^2 | 6.53×10^1 | 1.47×10^2 | 3.78×10^2 | 2.15×10^2 | 1.96×10^1 | 1.04×10^2 | 0 |
| | 排名 | 4 | 8 | 7 | 6 | 5 | 3 | 1 | 2 |
| F28 | 平均误差 | 1.03×10^3 | 3.90×10^3 | 1.92×10^3 | 2.15×10^3 | 1.43×10^3 | 3.49×10^3 | 1.94×10^3 | 2.00×10^2 |
| | 标准差 | 1.21×10^2 | 2.16×10^2 | 1.26×10^2 | 3.42×10^2 | 4.37×10^2 | 5.48×10^2 | 5.49×10^2 | 0 |
| | 排名 | 2 | 8 | 4 | 6 | 3 | 7 | 5 | 1 |
| F29 | 平均误差 | 1.40×10^3 | 3.59×10^4 | 2.00×10^4 | 3.32×10^3 | 3.08×10^6 | 5.44×10^5 | 1.98×10^7 | 2.00×10^2 |
| | 标准差 | 4.27×10^2 | 1.64×10^5 | 7.15×10^3 | 1.46×10^3 | 4.99×10^6 | 2.61×10^6 | 3.96×10^6 | 0 |
| | 排名 | 2 | 5 | 4 | 3 | 7 | 6 | 8 | 1 |
| F30 | 平均误差 | 4.63×10^3 | 1.55×10^4 | 1.97×10^4 | 1.61×10^4 | 6.47×10^3 | 2.49×10^4 | 6.96×10^6 | 2.00×10^2 |
| | 标准差 | 2.32×10^3 | 4.24×10^3 | 2.00×10^3 | 4.10×10^3 | 3.43×10^3 | 2.26×10^4 | 1.03×10^7 | 0 |
| | 排名 | 2 | 4 | 6 | 5 | 3 | 7 | 8 | 1 |
| | 平均排名 | 2.9333 | 7.0333 | 5.0333 | 4.7333 | 3.2667 | 5.1667 | 4.6333 | 3.0000 |
| | 最终排名 | 1 | 8 | 6 | 5 | 3 | 7 | 4 | 2 |

由表 6 可知, 与 HS-SA、CoDE 和 LM-BBO 算法相比, LIL-GWO 分别在 17、21 和 20 个测试函数上获得了较好的结果; 而 HS-SA、CoDE 和 LM-BBO 算法则分别在 13、9 和 10 个函数上得到了较好的实验结果. 与 MoABC、HSCA 和 DGS-TLBO 算法相比, LIL-GWO 算法分别在 20、18 和 22 个测试函数上取得了较好的结果, 在 F5、F24 和 F26 上获得了相似的结果. 然而, MoABC、HSCA 和 DGS-TLBO 算法分别在 9、11 和 7 个函数上得到了较好的结果. 与 PBA 算法相比, 除了函数 F3 和 F20, LIL-GWO 算法在其他函数上均获得了较好的结果; 对于 F3 和 F12, PBA 算法的结果要优. 由表 8 中 Friedman's 排名检验结果可知, HS-SA 排名第一, LIL-GWO 排名第二, 依次是 DGS-TLBO、LM-BBO、MoABC、CoDE、HSCA 和 PBA.

3.6 LIL-GWO 用于光伏模型参数辨识

为解决气候变化, 环境污染和经典化石燃料枯竭等问题, 近年来人们越来越关注可再生能源的使用. 太阳能因其广泛的实用性和清洁度而被认

为是最有前景的可再生能源之一^[52]. 它通过光伏 (Photo-voltaic, PV) 系统将太阳能转换为电能. 然而, PV 系统通常在恶劣的室外环境中运行, 这极大地影响了太阳能的利用效率. 为了控制和优化 PV 系统, 使用基于测量的电流-电压数据的精确模型来评估 PV 阵列在运行中的实际行为是非常重要的^[52]. 在众多描述 PV 系统的模型中, 单二极管模型在实践中被广泛使用^[52]. 然而, PV 模型的准确性主要取决于它们的参数. 因此, 准确可靠地辨识模型参数对 PV 系统的评估、优化和控制不可或缺.

如图 7 所示, 单二极管模型包括与二极管并联的电流源、分流电阻和串联电阻.

由图 7 可计算出输出电流 I_L 如下^[52]:

$$I_L = I_{ph} - I_d - I_{sh} = I_{ph} - I_{sd} \cdot \left[\exp \left(\frac{q \cdot (V_L + R_S \cdot I_L)}{n \cdot k \cdot T} \right) - 1 \right] - \frac{V_L + R_S \cdot I_L}{R_{sh}} \quad (20)$$

其中, I_{ph} 为光生电流, I_d 为二极管电流, I_{sh} 为并联

电阻电流, I_{sd} 为二极管的反向饱和电流, V_L 为输出电压, n 为二极管理想系数, $k = 1.3806503 \times 10^{-23}$ J/K 为玻尔兹曼常量, $q = 1.60217646$ 为电子电荷, T 为电池温度, R_S 和 R_{sh} 分别为串联和并联电阻. 由式 (21) 可知, 在单二极管模型中, 5 个未知参数 (I_{ph} , I_{sd} , R_S , R_{sh} 和 n) 需要辨识.

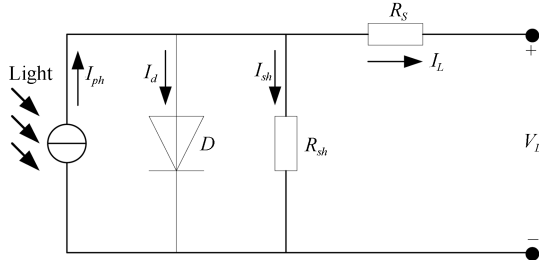


图 7 单二极管模型的结构

Fig. 7 Structure of single diode model

PV 模型的参数辨识可转化为优化问题, 其目标是使辨识参数得到的模拟数据与实测数据的误差最

小, 可用均方根误差 (RMSE) 来度量^[52]:

$$RMSE(\mathbf{x}) = \sqrt{\frac{1}{N} \sum_{k=1}^N [I_{L,calculated}^k(\mathbf{x}) - I_{L,measured}^k]^2}$$

其中, N 为数据的数量, $\mathbf{x} = (I_{ph}, I_{sd}, R_S, R_{sh}, n)$.

选取文献 [53] 中的实测数据, 利用 LIL-GWO 算法对单二极管 PV 模型的参数进行辨识. LIL-GWO 算法参数设置与第 3.1 节相同, 独立运行 30 次实验得到一组最优参数, 并与文献 [52] 中的几种算法进行比较, 结果如表 7 所示.

由表 7 中结果可知, LIL-GWO 辨识 PV 模型参数得到一组最优参数的 RMSE 值为 9.8604×10^{-4} . 与 BSA、IBSA、LBSA、CLPSO、BLPSO、DE-BBO 和 GOTLBO 算法相比, LIL-GWO 辨识最优参数后得到的 RMSE 值 (9.8604×10^{-4}) 最小. 此外, 利用 LIL-GWO 辨识的最优参数生成的电流-电压 (I-V) 和功率-电压 (P-V) 曲线如图 8 所示. 从图 8 可以清晰地看出, 在整个电压范围内, LIL-GWO 辨识最优参数后获得的模拟数据与实测数据高度吻合且具有较强的可靠性和鲁棒性.

表 7 不同算法对单二极管模型的最优辨识参数比较

Table 7 Comparison among different algorithms on single diode model

| 算法 | I_{ph} (A) | I_{sd} (μ A) | R_S (Ω) | R_{sh} (Ω) | n | RMSE |
|---------|--------------|---------------------|--------------------|-----------------------|--------|-------------------------|
| BSA | 0.7609 | 0.37749 | 0.0358 | 56.5266 | 1.4970 | 1.0398×10^{-3} |
| IBSA | 0.7607 | 0.35502 | 0.0361 | 58.2012 | 1.4907 | 1.0092×10^{-3} |
| LBSA | 0.7606 | 0.34618 | 0.0362 | 59.0978 | 1.4881 | 1.0143×10^{-3} |
| CLPSO | 0.7608 | 0.34302 | 0.0361 | 54.1965 | 1.4873 | 9.9633×10^{-3} |
| BLPSO | 0.7607 | 0.36620 | 0.0359 | 60.2845 | 1.4939 | 1.0272×10^{-3} |
| DE-BBO | 0.7605 | 0.32477 | 0.0364 | 55.2627 | 1.4817 | 9.9922×10^{-4} |
| GOTLBO | 0.7608 | 0.32970 | 0.0363 | 53.3664 | 1.4833 | 9.8856×10^{-3} |
| LIL-GWO | 0.7608 | 0.32363 | 0.0364 | 53.7967 | 1.4814 | 9.8604×10^{-4} |

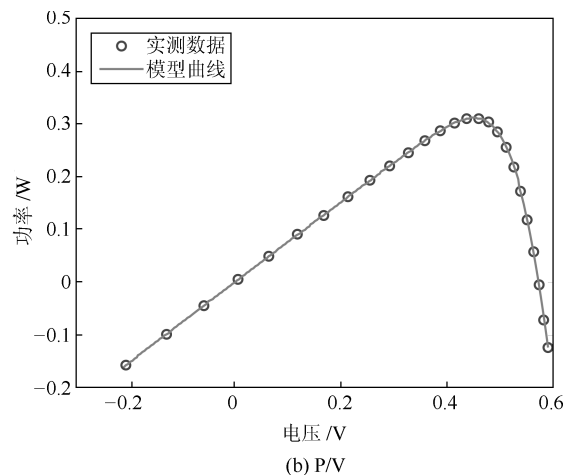
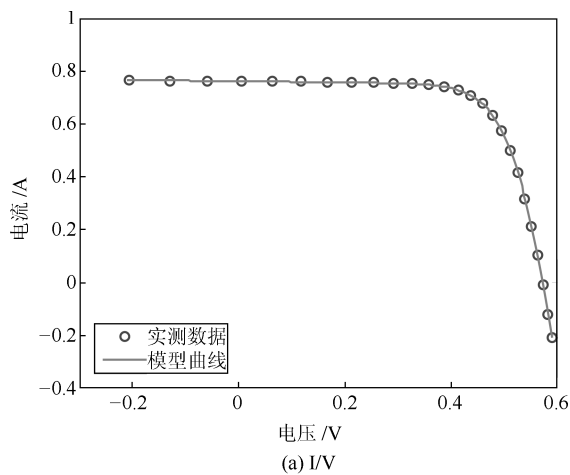


图 8 单二极管模型的测试数据和 LIL-GWO 计算数据的比较

Fig. 8 Comparison of the measured data and calculated data obtained by LIL-GWO for the single diode model

4 结论

本文提出一种新型的全局灰狼优化算法. 首先对 GWO 算法的控制参数 C 进行了一般性分析, 在此基础上提出一种修改控制参数 C 策略以协调算法的勘探和开采能力. 其次, 对群体在不同进化阶段的聚集程度进行了分析, 引入基于光透镜成像原理的反向学习策略, 扩大算法群体搜索的有效范围, 从而降低算法陷入局部最优的概率. 然而, 本文对算法失效的场景没有进行讨论.

第 1 组测试函数仿真实验表明, 与基本 GWO 算法、改进 GWO 算法和其他群体智能优化算法相比, LIL-GWO 在收敛速度和精度上均具有明显的优势, 在高维问题上也体现出较好的性能, 说明两个改进策略是有效的.

第 2 组是基于 CEC2014 的 30 个更复杂函数的测试, 与 7 种知名群体智能算法相比, LIL-GWO 也同样具有较强的竞争力.

太阳能光伏系统模型参数辨识问题的结果表明, 与所比较的辨识算法相比, LIL-GWO 算法获得最优辨识参数的 RMSE 值最小.

References

- 1 Lv Bai-Quan, Zhang Jing-Jing, Li Zhan-Pei, Liu Ting-Zhang. Fuzzy partical swarm optimization based on filled function and transformation function. *Acta Automatica Sinica*, 2018, **44**(1): 74–86
(吕柏权, 张静静, 李占培, 刘廷章. 基于变换函数与填充函数的模糊粒子群优化算法. *自动化学报*, 2018, **44**(1): 74–86)
- 2 Gao W F, Liu S Y, Huang L L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Transactions on Cybernetics*, 2013, **43**(3): 1011–1024
- 3 Xiao Hui-Hui, Wan Chang-Xuan, Duan Yan-Ming, Tan Qian-Lin. Flower pollination algorithm based on gravity search mechanism. *Acta Automatica Sinica*, 2017, **43**(4): 576–594
(肖辉辉, 万常选, 段艳明, 谭黔林. 基于引力搜索机制的花朵授粉算法. *自动化学报*, 2017, **43**(4): 576–594)
- 4 Zhou Xiao-Gen, Zhang Gui-Jun, Hao Xiao-Hu. Differential evolution algorithm with local abstract convex region partition. *Acta Automatica Sinica*, 2015, **41**(7): 1315–1327
(周晓根, 张贵军, 郝小虎. 局部抽象凸区域剖分差分进化算法. *自动化学报*, 2015, **41**(7): 1315–1327)
- 5 Rakhshani H, Rahati A. Snap-drift cuckoo search: A novel cuckoo search optimization algorithm. *Applied Soft Computing*, 2017, **52**: 771–794
- 6 Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, **95**: 51–67
- 7 Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, **69**: 46–61
- 8 Nuaekaew K, Artrit P, Pholdee N, Bureerat S. Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer. *Expert Systems with Applications*, 2017, **87**: 79–89
- 9 Precup R E, David R C, Petriu E M. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity. *IEEE Transactions on Industrial Electronics*, 2017, **64**(1): 527–534
- 10 Martin B, Marot J, Bourennane S. Mixed grey wolf optimizer for the joint denoising and unmixing of multispectral images. *Applied Soft Computing*, 2019, **74**: 385–410
- 11 Saxena A, Kumar R, Das S. β -chaotic map enabled grey wolf optimizer. *Applied Soft Computing*, 2019, **75**: 84–105
- 12 Medjahed S A, Ait Saadi T, Benyettou A, Ouali M. Gray wolf optimizer for hyperspectral band selection. *Applied Soft Computing*, 2016, **40**: 178–186
- 13 Yao Peng, Wang Hong-Lun. Three-dimensional path planning for UAV based on improved interfered fluid dynamical system and grey wolf optimizer. *Control and Decision*, 2016, **31**(4): 701–708
(姚鹏, 王宏伦. 基于改进流体扰动算法与灰狼优化的无人机三维航路规划. *控制与决策*, 2016, **31**(4): 701–708)
- 14 Mittal N, Singh U, Sohi B S. Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, 2016, **2016**: 7950348
- 15 Rodríguez L, Castillo O, Soria J. Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic. In: *Proceedings of the 2016 IEEE Congress on Evolutionary Computation*. Vancouver, Canada: IEEE, 2016. 3116–3123
- 16 Long W, Jiao J J, Liang X M, Tang M Z. Inspired grey wolf optimizer for solving large-scale function optimization problems. *Applied Mathematical Modelling*, 2018, **60**: 112–126
- 17 Wei Zheng-Lei, Zhao Hui, Li Mu-Dong, Wang Yuan, Ke Yi-Ming. A grey wolf optimization algorithm based on non-linear 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, 王渊, 柯益明. 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)
- 18 Long W, Jiao J J, Liang X M, Tang M Z. An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, 2018, **68**: 63–80
- 19 Rodríguez L, Castillo O, Soria J, Melin P, Valdez F, Gonzalez C I, et al. A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Computing*, 2017, **57**: 315–328
- 20 Jaiswal K, Mittal H, Kukreja S. Randomized grey wolf optimizer (RGWO) with randomly weighted coefficients. In: *Proceedings of the 10th International Conference on Contemporary Computing*. Noida, India: IEEE, 2017. 1–3
- 21 Malik M R S, Mohideen E R, Ali L. Weighted distance grey wolf optimizer for global optimization problems. In: *Proceedings of the 2015 IEEE International Conference on Computational Intelligence and Computing Research*. Madurai, India: IEEE, 2015. 1–6

- 22 Wei Zheng-Lei, Zhao Hui, Han Bang-Jie, Sun Chu, Li Mu-Dong. Grey wolf optimization algorithm with self-adaptive searching strategy. *Computer Science*, 2017, **44**(3): 259–263 (魏政磊, 赵辉, 韩邦杰, 孙楚, 李牧东. 具有自适应搜索策略的灰狼优化算法. *计算机科学*, 2017, **44**(3): 259–263)
- 23 Tawhid M A, Ali A F. A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. *Memetic Computing*, 2017, **9**(4): 347–359
- 24 Kamboj V K. A novel hybrid PSO-GWO approach for unit commitment problem. *Neural Computing and Applications*, 2016, **27**(6): 1643–1655
- 25 Arora S, Singh H, Sharma M, Sharma S, Anand P. A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access*, 2019, **7**: 26343–26361
- 26 Daniel E, Anitha J, Gnanaraj J. Optimum Laplacian wavelet mask based medical image using hybrid cuckoo search-grey wolf optimization algorithm. *Knowledge-based Systems*, 2017, **131**: 58–69
- 27 Zhang X M, Kang Q, Cheng J F, Wang X. A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer. *Applied Soft Computing*, 2018, **67**: 197–214
- 28 Long Wen, Zhao Dong-Quan, Xu Song-Jin. Improved grey wolf optimization algorithm for constrained optimization problem. *Journal of Computer Applications*, 2015, **35**(9): 2590–2595 (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. *计算机应用*, 2015, **35**(9): 2590–2595)
- 29 Mahdad B, Srairi K. Blackout risk prevention in a smart grid based flexible optimal strategy using grey wolf-pattern search algorithms. *Energy Conversion and Management*, 2015, **98**: 411–429
- 30 Lu C, Gao L, Yi J. Grey wolf optimizer with cellular topological structure. *Expert Systems with Applications*, 2018, **107**: 89–114
- 31 Heidari A A, Pahlavani P. An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing*, 2017, **60**: 115–134
- 32 Gupta S, Deep K. A novel random walk grey wolf optimizer. *Swarm and Evolutionary Computation*, 2019, **44**: 101–112
- 33 Wolpert D H, Macready W G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997, **1**(1): 67–82
- 34 Kumar V, Kumar D. An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems. *Advances in Engineering Software*, 2017, **112**: 231–254
- 35 Rahnamayan S, Tizhoosh H R, Salama M M A. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation*, 2008, **12**(1): 64–79
- 36 Yu Fei, Li Yuan-Xiang, Wei Bo, Xu Xing, Zhao Zhi-Yong. The application of a novel OBL based on lens imaging principle in PSO. *Acta Electronica Sinica*, 2014, **42**(2): 230–235 (喻飞, 李元香, 魏波, 徐星, 赵志勇. 透镜成像反学习策略在粒子群算法中的应用. *电子学报*, 2014, **42**(2): 230–235)
- 37 Wang Hui. Research on intelligent algorithms based on region transformation search and its applications [Ph. D. dissertation], Wuhan University, China, 2011 (王晖. 区域变换搜索的智能算法研究. [博士学位论文], 武汉大学, 中国, 2011)
- 38 Hansen N, Müller S D, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 2004, **11**(1): 1–18
- 39 Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computer & Operations Research*, 2006, **33**(3): 859–871
- 40 Zhu G P, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 2010, **217**(7): 3166–3173
- 41 Rao R V, Patel V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *Journal of Industrial Engineering Computations*, 2012, **3**(4): 535–560
- 42 Hu H, Bai Y, Xu T. Improved whale optimization algorithm based on inertia weights and theirs applications. *International Journal of Circuits, Systems and Signal Processing*, 2017, **11**: 12–26
- 43 Xu Song-Jin, Long Wen. Improved sine cosine algorithm for solving high-dimensional optimization problems. *Application Research of Computers*, 2018, **35**(9): 2574–2577 (徐松金, 龙文. 求解高维优化问题的改进正弦余弦算法. *计算机应用研究*, 2018, **35**(9): 2574–2577)
- 44 Liang J J, Qu B Y, Suganthan P N. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Technical Report 201311, Nanyang Technological University, Singapore, 2014.
- 45 Assad A, Deep K. A hybrid harmony search and simulated annealing algorithm for continuous optimization. *Information Sciences*, 2018, **450**: 246–266
- 46 Kallioras N A, Lagaros N D, Avtzis D N. Pity beetle algorithm — a new metaheuristic inspired by the behavior of bark beetles. *Advances in Engineering Software*, 2018, **121**: 147–166
- 47 Wang Y, Cai Z X, Zhang Q F. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 2011, **15**(1): 55–66
- 48 Akbari R, Hedayatzadeh R, Ziarati K, Hassanizadeh B. A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation*, 2012, **2**: 39–52
- 49 Zou F, Wang L, Hei X H, Chen D B, Yang D D. Teaching-learning-based optimization with dynamic group strategy for global optimization. *Information Sciences*, 2014, **273**: 112–131
- 50 Mlakar U, Fister Jr I, Fister I. Hybrid self-adaptive cuckoo search for global optimization. *Swarm and Evolutionary Computation*, 2016, **29**: 47–72

- 51 Garg V, Deep K. Performance of Laplacian biogeography-based optimization algorithm on CEC 2014 continuous optimization benchmarks and camera calibration problem. *Swarm and Evolutionary Computation*, 2016, **27**: 132–144
- 52 Yu K J, Liang J J, Qu B Y, Cheng Z P, Wang H S. Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models. *Applied Energy*, 2018, **226**: 408–422
- 53 Easwarakhanthan T, Bottin J, Bouhouch I, Boutrif C. Non-linear minimization algorithm for determining the solar cell parameters with microcomputers. *International Journal of Solar Energy*, 1986, **4**: 1–12



龙文 贵州财经大学教授。2011年获得中南大学信息科学与工程学院控制科学与工程专业博士学位。主要研究方向为智能优化算法, 数据挖掘及应用。

E-mail: longwen227@mail.gufe.edu.cn

(**LONG Wen** Professor at Guizhou University of Finance and Economics. He received his Ph. D. degree in control

science and engineering from the School of Information Science and Engineering, Central South University in 2011. His research interest covers intelligence optimization algorithm, data mining and their applications.)



伍铁斌 湖南人文科技学院教授。2014年获得中南大学信息科学与工程学院控制科学与工程专业博士学位。主要研究方向为智能优化算法, 复杂生产过程建模与控制。

E-mail: wutiebin81@csu.edu.cn

(**WU Tie-Bin** Professor at Hunan University of Humanities, Science and

Technology. He received his Ph. D. degree in control science and engineering from the School of Information Science and Engineering, Central South University in 2014. His research interest covers intelligence optimization algorithm, complex production process modeling and control.)



唐明珠 长沙理工大学副教授。2011年获得中南大学信息科学与工程学院控制科学与工程专业博士学位。主要研究方向为故障诊断, 机器学习。

E-mail: tmz@csust.edu.cn

(**TANG Ming-Zhu** Associate professor at Changsha University of Science and Technology. He received his

Ph. D. degree in control science and engineering from the School of Information Science and Engineering, Central South University in 2011. His research interest covers fault diagnosis and machine learning.)



徐明 贵州财经大学教授。2016年获得云南大学物理科学与技术学院系统分析与集成专业博士学位。主要研究方向为智能优化算法, 数据挖掘及应用。

E-mail: xuming@mail.gufe.aedu.cn

(**XU Ming** Professor at Guizhou University of Finance and Economics. He received his Ph. D. degree in system

analysis and integration from the School of Physical Science and Technology, Yunnan University in 2016. His research interest covers intelligence optimization algorithm, data mining and their applications.)



蔡绍洪 贵州财经大学教授。主要研究方向为大数据挖掘, 智能优化算法, 复杂系统分析及应用。本文通信作者。

E-mail: caish@mail.gufe.edu.cn

(**CAI Shao-Hong** Professor at Guizhou University of Finance and Economics. His research interest covers big data mining, intelligence op-

timization algorithm, complex system analysis and their applications. Corresponding author of this paper.)