文章编号：2095-2163(2024)03-0046-08　　中图分类号：TE341　　文献标志码：A

一种求解最优潮流的改进灰狼优化算法

王　恒，杨　婷

（铜仁职业技术学院 信息工程学院，贵州 铜仁 554300）

摘　要：最优潮流是电力系统最关键的问题之一，本文采用一种求解最优潮流的改进灰狼优化算法（LMGWO）求解最优潮流（OPF）问题，该算法引入算术优化算法（Arithmetic Optimization Algorithm，AOA）中的乘除算子，利用带透镜成像的反向学习策略增强最优个体的多样性，提高算法跳出局部最优的能力。通过与几种常用的算法进行对比实验表明：本文提出的 LWGWO 算法是有竞争力的，总体上优于对比算法；LMGWO 算法在最小化燃料成本、有功输电损耗和改善电压偏差方面更有效地找到了最优潮流（OPF）问题的最优解。

关键词：灰狼优化算法；最优潮流；算术优化算法；燃料成本；有功输电损耗

An improved grey wolf optimization algorithm for solving optimal power flow

WANG Heng，YANG Ting

（School of Information Engineering，Tongren Polytechnic College，Tongren 554300，Guizhou，China）

Abstract：Optimal power flow is one of the most critical problems in power system. In this paper，an improved Grey Wolf Optimization Algorithm（LMGWO）is used to solve the optimal power flow（OPF）problem. In this algorithm，multiplication and division operators in the Arithmetic Optimization Algorithm（AOA）are introduced. The reverse learning strategy with lens imaging is used to enhance the diversity of optimal individuals and improve the ability of the algorithm to jump out of the local optimal. Through comparative experimental analysis of several commonly used algorithms，the proposed LWGWO algorithm is competitive and generally superior to recent algorithms. The experimental results show that LMGWO algorithm can find the optimal solution of OPF problem more effectively in terms of minimizing fuel cost，active power transmission loss and improving voltage deviation.

Key words：grey wolf optimization algorithm；optimal power flow；arithmetic optimization algorithm；fuel cost；active power transmission loss

0　引　言

最优潮流（OPF）问题是电力系统运行过程中备受关注的焦点问题，旨在找到最优的运行方式，使得电力系统的运行成本最低，同时满足安全、稳定和环保等约束条件。OPF 问题的求解是在满足一系列物理、环境、实际和运行的约束条件下，通过优化特定的目标来确定电力系统的运行状态。

在此之前，许多传统的优化技术的应用已获成功，包括基于梯度的方法、牛顿法、单纯形法、序列线性规划和内点法[1-5]。由于 OPF 问题本质上是一个多极、多约束、非凸的复杂优化问题，使用传统的数值方法来求解，过程复杂、耗时且精度较差。近年来，元启发式算法的快速发展为解决 OPF 问题提供了更多的选择。元启发式算法具有参数少、易于操作、不需要梯度信息等优点，能够在合理的时间内和高度复杂的约束条件下找到复杂问题的最优解。刘自发等学者[6]提出了一种基于混沌粒子群优化方法的电力系统无功最优潮流（OPF）问题。Farhat 等学者[7]提出了一种基于邻域维度学习搜索策略的增强型黏液霉菌算法（enhanced slime mould algorithm，ESMA）用于求解最优潮流（OPF）问题等等。越来越多的元启发式算法被广泛用于解决电力系统优化相关问题[8-13]。

灰狼优化算法（grey wolf optimizer，GWO）是由 Mirjalili 等学者[14]在 2014 年上提出的一种新的元启发式算法。灰狼优化算法（GWO）原理简单、编程容易、需要调整的参数少，现已陆续应用于电力系

基金项目：铜仁市科学技术局基础科学研究项目（铜市科研（2022）72 号）。

作者简介：王　恒（1985-），男，博士研究生，讲师，主要研究方向：智能计算与混合系统、人工智能、故障诊断研究等。Email：wangheng_trzy@foxmail.com

收稿日期：2023-06-16

统、自动控制、能源市场战略招标等领域[15-17]。然而,与许多元启发式优化算法一样,灰狼优化算法(GWO)在求解复杂的非线性问题时容易陷入局部最优且收敛速度慢。

针对原有灰狼优化算法在求解最优潮流(OPF)问题时存在的不足,提出了一种改进的灰狼优化算法(LMGWO算法)。基于镜头成像学习和乘除算子策略对原灰狼优化算法(GWO)进行改进,主要有2点改进:

(1)为了增强算法的全局探索能力,引入乘除算子策略,提高算法的收敛速度;

(2)为增强最优个体的多样性,引入透镜成像修正反向学习策略,提高算法跳出局部最优的能力。

1　最优潮流公式

最优潮流(OPF)问题是典型的多变量、多约束的非线性组合优化问题。最优潮流(OPF)问题的求解过程是通过寻找最优的控制变量来获得最小的目标函数。数学模型定义如下:

$$\min F(u,x)$$
$$\text{s.t.}\begin{cases} g(u,x)=0 \\ h(u,x)\leqslant 0 \end{cases}$$

其中,F 表示目标函数;x 表示控制变量;u 表示状态变量;$g(u,x)=0$ 是等式约束;$h(u,x)\leqslant 0$ 是不等式约束。

1.1　控制变量和状态变量

最优潮流(OPF)问题公式中的控制变量集合为:

$$x=[P_{G_2},\cdots,P_{G_{NG}},V_{G_1},\cdots,V_{G_{NG}},T_1,\cdots,T_{NT},Q_{C_1},\cdots,Q_{C_{NC}}] \tag{1}$$

其中,$P_{G_2},\cdots,P_{G_{NG}}$ 为系统除松弛母线外的有功发电量;$V_{G_1},\cdots,V_{G_{NG}}$ 为系统的电压幅值;$T_1,\cdots,T_{NT}$ 为变压器分接设定值;$Q_{C_1},\cdots,Q_{C_{NC}}$ 为并联无功补偿;NG、NT、NC 分别为发电机个数、调节变压器个数、无功补偿器个数。

最优潮流(OPF)问题表述的状态变量集合为:

$$u=[P_{G_1},V_{L_1},\cdots,V_{L_{NL}},Q_{G_1},\cdots,Q_{G_{NG}},S_{l_1},\cdots,S_{l_{nl}}] \tag{2}$$

其中,P_G 为空闲母线输出有功功率;V_L 为负载母线电压幅值;Q_G 为各发电机组输出无功功率;S_l 为输电线路负载。

1.2　目标函数

将燃油成本、有源输电损耗和电压偏差作为最优潮流(OPF)问题的目标函数。各目标函数的数学模型定义如下。

(1)燃料成本(FC)。描述发电成本的目标函数,可得数学建模如下:

$$F_1(x,u)=\sum_{i=1}^{N_g}(a_i+b_iP_{Gi}+c_iP_{Gi}^2) \tag{3}$$

其中,N_g 为发电机个数;a_i, b_i, c_i 为第 i 台发电机组的燃料成本系数;P_{Gi} 为第 i 台发电机组的实际发电量。

(2)有功输电损耗(APL)。传输线的 APL 可表示为:

$$F_2(x,u)=\sum_{i,j\in N_l}G_{ij}(V_i^2+V_j^2-2V_iV_j\cos(\theta_{ij})) \tag{4}$$

其中,N_l 为输电线路数;G_{ij} 为线路 ij 的传递电导;V_i 为第 i 根母线的电压幅值;V_j 为第 j 根母线的电压幅值;θ_{ij} 为母线 i 与 j 之间的电压相角之差。

1.3　约束条件

在最优潮流(OPF)问题中,等式约束和不等式约束是电力系统需要满足的约束,通常是每个节点的功率平衡约束,可以通过式(5)和式(6)进行定义:

$$P_{Gi}-P_{Di}=V_i\sum_{i,j=1}^{N}V_j(G_{ij}\cos(\delta_i-\delta_j)+B_{ij}\sin(\delta_i-\delta_j)) \tag{5}$$

$$Q_{Gi}-Q_{Di}=V_i\sum_{i,j=1}^{N}V_j(G_{ij}\sin(\delta_i-\delta_j)-B_{ij}\cos(\delta_i-\delta_j)) \tag{6}$$

其中,P_{Di}、Q_{Di} 分别为第 i 台母线的有功、无功功率;P_{Gi} 和 Q_{Gi} 为第 i 台发电机的无功发电量;N 为母线个数;G_{ij} 和 B_{ij} 分别为母线 i 和 j 之间的电导和电纳;V_i 和 V_j 分别为母线 i 和 j 的电压幅值。

2　改进的灰狼优化算法

2.1　灰狼优化算法

灰狼优化算法(GWO)是模仿自然界灰狼群体社会等级和捕食行为而衍生的一种元启发式算法[14]。灰狼群体的社会等级为 α 狼、β 狼、δ 狼和 ω 狼。狼的狩猎行为分为跟踪、包围和攻击猎物三个步骤。狼群包围猎物的数学模型定义为:

$$X=X_\alpha(t)-A\cdot|C\cdot X_\alpha(t)-X(t)| \tag{7}$$

其中,X 和 X_α 分别表示狼个体和猎物个体的位置向量,t 表示当前迭代次数。

系数向量 A 和 C 定义为:

$$A=2a\cdot r_1-a \tag{8}$$

$$C=2\cdot r_2 \tag{9}$$

其中,r_1 和 r_2 是[0,1]之间的随机向量,a 从2线性递减到0,其数学模型定义为:

$$a = 2 - \frac{2 \cdot t}{T_{\max}} \tag{10}$$

其中，$T_{\max}$ 为最大迭代次数。

包围猎物后，β 狼和 δ 狼在 α 狼的带领下追捕猎物。在追捕过程中，狼群的个体位置会随着猎物的逃跑而发生变化。因此，灰狼群可以根据 α、β、δ 的位置 X_α，X_β，X_δ 更新灰狼的位置：

$$X_1 = X_\alpha(t) - A_1 \cdot | C_1 \cdot X_\alpha(t) - X(t) | \tag{11}$$

$$X_2 = X_\beta(t) - A_2 \cdot | C_2 \cdot X_\beta(t) - X(t) | \tag{12}$$

$$X_3 = X_\delta(t) - A_3 \cdot | C_3 \cdot X_\delta(t) - X(t) | \tag{13}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{14}$$

其中，$X(t+1)$ 是当前个体的位置。

2.2　改进 GWO 算法的思路和策略

2.2.1　算术乘除运算符策略

2021年，Abualigah 等学者[18]提出的一种新的元启发式算法，即算术优化算法（Arithmetic Optimization Algorithm，AOA），主要利用数学中的乘、除运算符以及加、减运算符四种混合运算。AOA 中的乘除算子具有较强的全局探索能力。灰狼种群在更新位置时侧重使用 α 狼、β 狼和 δ 狼作为精英来引导搜索，具有较强的局部开发能力。引入算术乘除算子策略，提高 GWO 算法的全局探索能力。算术乘除算子策略的数学模型定义为：

$$X_i^j(t+1) = \begin{cases} X_{\text{best}}^j \div (MOP + \varepsilon) \cdot [(ub_j - lb_j) \cdot \mu + lb_j], & r_3 \leqslant 0.5 \\ X_{\text{best}}^j \times MOP \cdot [(ub_j - lb_j) \cdot \mu + lb_j], & r_3 > 0.5 \end{cases} \tag{15}$$

其中，X_{best}^j 表示当前最优解的第 j 个位置；r_3 表示介于[0,1]之间的随机数；ε 表示防止分母为0的整数；μ 表示调节搜索过程的控制参数，μ 的值在基本 AOA 中为0.5；ub_j 和 lb_j 分别表示第 i 个位置的上下界。MOP 为概率函数，其数学模型描述为：

$$MOP = 1 - \frac{t^{\frac{1}{\tau}}}{T_{\max}^{\frac{1}{\tau}}} \tag{16}$$

其中，$\tau = 5$ 是一个敏感因子，定义了迭代的搜索精度。

由式(15)可知，AOA 可以带来高分布，借助乘除算子实现位置更新，可以大大提高算法的全局探索能力。本文设置阈值为0.3。

2.2.2　基于透镜成像的反向学习策略

根据灰狼的位置更新公式，由 α 狼、β 狼和 δ 狼带领群体中的其他狼进行位置更新。如果 α 狼、β 狼和 δ 狼都处于局部最优，则整个群体会聚集在局部最优区域，导致种群陷入局部最优。针对该问题，本文提出一种基于透镜成像原理的反向学习方法，将对立个体与当前最优个体相结合，生成新个体。

假设在一维空间中，在轴区间[lb, ub]上有一个高度为 H 的个体 P，其在 x 轴上的投影为 X（X 为全局最优个体）。将焦距为 F 的镜头放置在基点位置 O 上（本文取基点位置为 $(lb + ub/2)$）。个体 P 通过透镜，以获得高度为 H 的倒置图像 P^*，在这点上，第一个倒置的个体 x 通过透镜成像在 X 轴上产生。镜头图像的反向学习策略如图1所示。

在图1中，全局最优个体 X 以 O 为基点找到其对应的逆个体 X^*。因此，可以从透镜成像原理推导出数学模型，推得的公式为：

$$\frac{(ub + lb)/2 - X}{X^* - (ub + lb)/2} = \frac{h}{h^*} \tag{17}$$

设 $h/h^* = k$，k 表示拉伸因子。通过推导式(17)，可以得到反转点 X^* 的计算公式：

$$X^* = \frac{ub + lb}{2} + \frac{ub + lb}{2k} - \frac{X}{k} \tag{18}$$

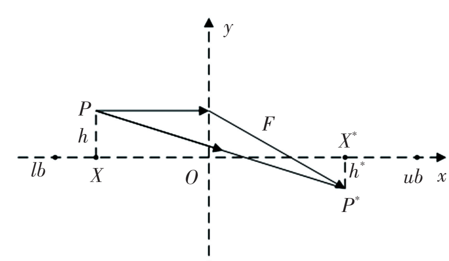


图1　基于镜头图像的反向学习策略

Fig. 1　Reverse learning strategy based on lens image

在算法搜索解时，使用拉伸因子 k 作为微观调节因子，增强算法的局部开发能力。然而，在基本的透镜成像逆学习策略中，拉伸因子一般作为固定值使用，不允许算法探索解空间的全范围。为此，本文提出一种基于非线性动态递减的伸缩因子策略，在算法迭代初期可以得到较大的值，有助于算法在不同维度的区域进行更大范围的搜索，以提高种群的多样性。非线性动态拉伸因子定义为：

$$k = k_{\max} - (k_{\max} - k_{\min}) \cdot \left[1 - \cos\left(\frac{\pi t}{2T_{\max}}\right)\right] \tag{19}$$

其中，$k_{\max}$ 和 $k_{\min}$ 分别表示最大和最小拉伸因子，$T_{\max}$ 表示最大迭代次数。

可以将式(18)扩展到 D - 维搜索空间，得到数学模型为：

$$X_j^* = \frac{ub_j + lb_j}{2} + \frac{ub_j + lb_j}{2k} - \frac{X_j}{k} \quad (20)$$

其中，X_j 和 X_j^* 分别表示 X 和 X^* 的的第 j 维向量，ub_j 和 lb_j 分别表示决策变量的第 j 维向量。

基于透镜的反向学习策略虽然极大地提高了算法的求解精度，但无法直接判断生成的新反向个体是否优于原始个体。因此，本文引入贪心机制来比较新旧个体适应度值，从而筛选出最优个体。该方法不断获得更好的解，提高了算法的寻优能力。贪婪机制的数学模型描述如下：

$$X_{new}(t) = \begin{cases} X^*, & f(X) > f(X^*) \\ X, & f(X) \leqslant f(X^*) \end{cases} \quad (21)$$

2.2.3　LMGWO 算法实现过程

LMGWO 算法实现流程如图 2 所示。

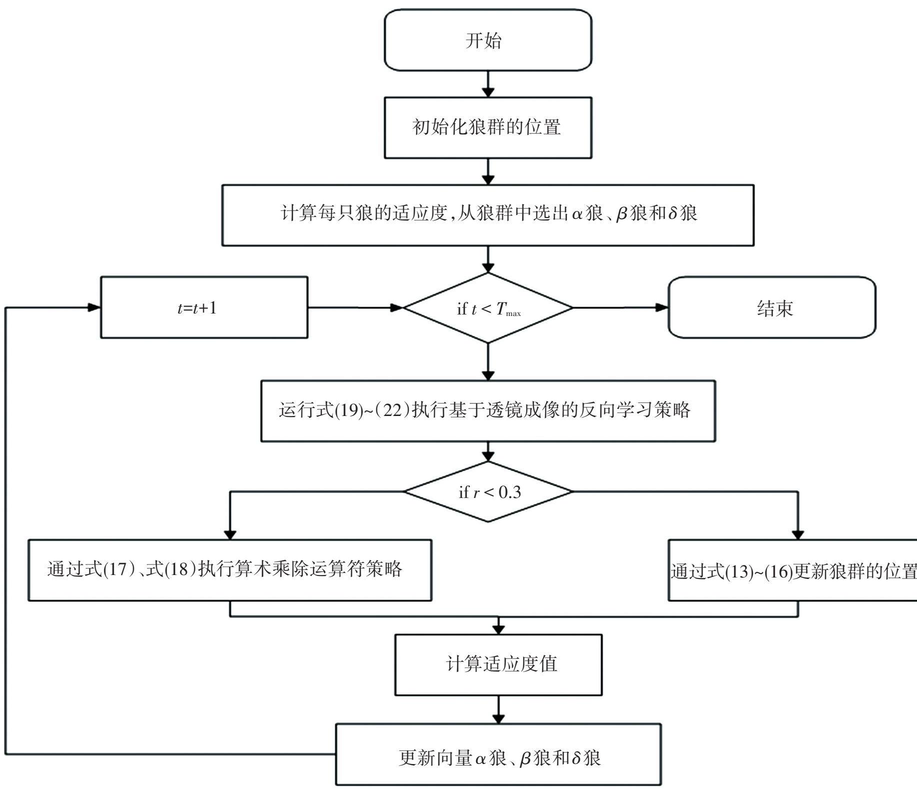


图 2　LMGWO 算法流程图

Fig. 2　Flow chart of LMGWO algorithm

3　实验

3.1　实验环境及参数设置

在 Intel（R）Core（TM）i7－i7－6500U CPU、2.50 GHz频率、8 GB 内存、Windows 10（64 bit）操作系统上进行仿真实验，编程软件为 Matlab R2018a。采用 9 个基准测试函数，包括 5 个单峰函数 $F_1 \sim F_5$ 和 4 个非线性多峰函数 $F_6 \sim F_9$，见表 1。参与对比的灰狼优化算法（GWO）[14]、算术优化算法（AOA）[18]、正弦余弦算法（SCA）[19]、猩猩优化算法（ChOA）[20]、鲸鱼优化算法（WOA）[21]、LMGWO 的参数设置见表 2。

表 1　基准测试函数

Table 1 Benchmark functions

函数编号	名称	维度	范围	最优值
F_1	*Sphere*	30	[－100，100]	0
F_2	*Schwefel.2.22*	30	[－10，10]	0
F_3	*Schwefel.1.2*	30	[－100，100]	0
F_4	*Schwefel.2.21*	30	[－100，100]	0
F_5	*Quartic*	30	[－1.28，1.28]	0
F_6	*Rastrigin*	30	[－5.12，5.12]	0
F_7	*Ackley*	30	[－32，32]	0
F_8	*Criewank*	30	[－600，600]	0
F_9	*Apline*	30	[－10，10]	0

表 2 算法参数设置

Table 2 Parameter settings of algorithms

算法名称	参数设置
SCA [19]	$M = 2$
ChOA [20]	$f_{max} = 2.5, f_{min} = 0$
WOA [21]	$a_{max} = 2, a_{min} = 0, b = 1$
AOA [18]	$MOP\_Max = 1, MOP\_Min = 0.2, \alpha = 5, \mu = 0.499$
GWO [14]	$a_{max} = 2, a_{min} = 0$
LMGWO	$a_{max} = 2, a_{min} = 0$

3.2 算法性能对比分析

为了验证了 LMGWO 算法的有效性和优越性，将 LMGWO 算法与灰狼优化算法(GWO)[14]、算术优化算法(AOA)[18]、正弦余弦算法(SCA)[19]、猩猩优化算法(ChOA)[20]、鲸鱼优化算法(WOA)[21]在 9 个不同特性的基准测试函数上进行仿真实验。在各个算法的测试环境相同的条件下，种群规模 $N = 30$，空间维度 $Dim = 30$，最大迭代次数 $T_{max} = 500$。采用均值和标准差作为实验的评价指标，均值和标准差越小，表明算法的性能越好。6 种算法对 9 个基准函数的求解结果见表 3。

表 3 各算法在基准函数上的优化性能比较

Table 3 Optimization performance comparison of each algorithm on the benchmark function

函数编号	指标	SCA	ChOA	WOA	AOA	GWO	LMGWO
F_1	*Mean* 均值	2.82×10^{1}	5.45×10^{-6}	2.20×10^{-72}	1.57×10^{-7}	1.84×10^{-27}	0
	Std 标准差	7.15×10^{1}	3.34×10^{-6}	1.34×10^{-71}	4.36×10^{-7}	2.35×10^{-28}	0
F_2	*Mean* 均值	6.48×10^{-2}	5.48×10^{-5}	5.55×10^{-51}	4.08	1.02×10^{-16}	0
	Std 标准差	3.45×10^{-2}	5.02×10^{-5}	9.54×10^{-51}	5.11	4.61×10^{-17}	0
F_3	*Mean* 均值	1.25×10^{4}	6.45×10^{2}	1.02×10^{4}	9.61×10^{3}	5.21×10^{-5}	0
	Std 标准差	3.16×10^{3}	8.64×10^{2}	6.32×10^{4}	3.22×10^{2}	1.17×10^{-4}	0
F_4	*Mean* 均值	2.77×10^{1}	9.15×10^{-1}	4.11×10^{1}	1.21	1.04×10^{-6}	0
	Std 标准差	5.68×10^{1}	5.47×10^{-1}	2.19×10^{1}	1.39	1.47×10^{-6}	0
F_5	*Mean* 均值	3.27×10^{-2}	7.64×10^{-3}	2.45×10^{-3}	5.13×10^{-1}	2.30×10^{-3}	2.45×10^{-5}
	Std 标准差	5.98×10^{-2}	5.16×10^{-3}	3.09×10^{-3}	3.18×10^{-2}	1.70×10^{-3}	2.04×10^{-5}
F_6	*Mean* 均值	3.02×10^{1}	8.99×10^{1}	6.11×10^{-15}	4.67×10^{1}	4.28	0
	Std 标准差	6.48×10^{1}	1.02×10^{1}	1.98×10^{-14}	2.13×10^{1}	5.44	0
F_7	*Mean* 均值	5.51	4.07×10^{1}	1.11×10^{-15}	2.45×10^{-1}	2.05×10^{-13}	8.88×10^{-16}
	Std 标准差	1.84	5.11×10^{-2}	7.16×10^{-15}	4.41	1.17×10^{-14}	0
F_8	*Mean* 均值	3.65	3.47×10^{-2}	6.39×10^{-2}	2.58×10^{-2}	4.68×10^{-3}	0
	Std 标准差	2.00×10^{-1}	5.19×10^{-2}	4.77×10^{-2}	8.12×10^{-2}	7.55×10^{-3}	0
F_9	*Mean* 均值	4.55×10^{-2}	5.40×10^{-3}	5.49×10^{-39}	4.11×10	6.79×10^{-4}	0
	Std 标准差	1.36×10^{-2}	1.24×10^{-2}	2.33×10^{-38}	2.28×10	1.17×10^{-4}	0

由表 3 可以看出，在基准测试中，对于 $F_1 \sim F_4$、F_6、F_8 和 F_9 函数，对比算法均未能找到最优解，而 LMGWO 算法达到 100%的求解精度。在求解 F_5 和 F_8 函数时，LMGWO 的求解精度优于其他 5 种对比算法，但也与其他算法一样容易陷入局部最优。基于以上分析说明 LMGWO 算法比其他算法具有更高的求解精度和稳定性，证明了其有效性和优越性。

3.3 LMGWO 算法在高维条件的性能分析

为了进一步验证 LMGWO 求解高维优化问题的性能，以算法解的均值和平均变化率为评价指标，对 9 个函数在 100~500 维增量下进行测试，将本文提出的 LMGWO 算法与原始 GWO 算法独立运行 30 次，并记录其均值，实验结果见表 4。

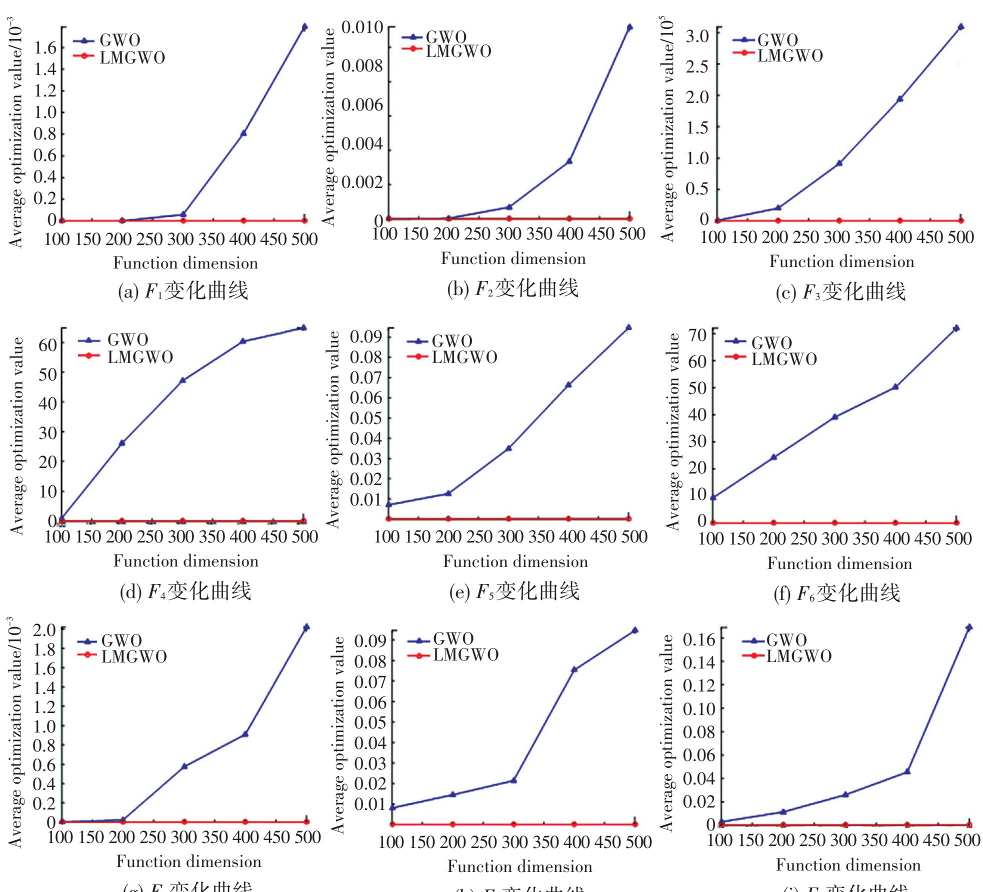
由表 4 可知，随着维数的增加，LMGWO 的均值基本保持不变，F_1、F_2、F_3、F_4、F_6、F_9 函数的 LMGWO 均值保持为 0。随着维数的增加，GWO 均值呈现增加趋势。在测试函数 F_5 上，LMGWO 算法的均值基本保持不变，而 GWO 算法的均值变化明显大于 LMGWO 算法；在测试函数 F_8 上，LMGWO 算法的平均变化率均为 0，远低于 GWO 算法的平均变化率。

表 4　LMGWO 与 GWO 在不同维度下优化函数均值的比较

Table 4　Comparison of LMGWO and GWO optimization function mean values in different dimensions

函数编号	算法名称	维数					平均变化率/%
		100	200	300	400	500	
F_1	GWO	1.46×10^{-12}	1.43×10^{-7}	5.79×10^{-5}	8.08×10^{-4}	1.79×10^{-3}	4.48×10^{-4}
	LMGWO	0	0	0	0	0	0
F_2	GWO	5.35×10^{-8}	3.25×10^{-5}	6.79×10^{-4}	3.34×10^{-3}	1.12×10^{-2}	2.80×10^{-3}
	LMGWO	0	0	0	0	0	0
F_3	GWO	7.31×10^{2}	2.02×10^{4}	9.11×10^{4}	1.94×10^{5}	3.09×10^{5}	7.71×10^{4}
	LMGWO	0	0	0	0	0	0
F_4	GWO	8.82×10^{-1}	2.61×10^{1}	4.71×10^{1}	6.03×10^{1}	6.48×10^{1}	1.60×10^{1}
	LMGWO	0	0	0	0	0	0
F_5	GWO	7.03×10^{-3}	1.26×10^{-2}	3.49×10^{-2}	6.63×10^{-2}	9.46×10^{-2}	2.19×10^{-2}
	LMGWO	3.41×10^{-5}	3.87×10^{-5}	4.05×10^{-5}	4.72×10^{-5}	6.39×10^{-5}	7.45×10^{-6}
F_6	GWO	9.29	2.42×10^{1}	3.91×10^{1}	5.02×10^{1}	7.20×10^{1}	1.57×10^{1}
	LMGWO	0	0	0	0	0	0
F_7	GWO	6.77×10^{-7}	2.22×10^{-5}	5.74×10^{-4}	9.09×10^{-4}	2.02×10^{-3}	5.05×10^{-4}
	LMGWO	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	0
F_8	GWO	8.05×10^{-3}	1.45×10^{-2}	2.14×10^{-2}	7.53×10^{-2}	9.46×10^{-2}	2.16×10^{-2}
	LMGWO	0	0	0	0	0	0
F_9	GWO	2.81×10^{-3}	1.13×10^{-2}	2.59×10^{-2}	4.54×10^{-2}	1.69×10^{-1}	4.15×10^{-2}
	LMGWO	0	0	0	0	0	0

2 种算法在不同维度下均值的变化情况如图 3 所示。在 9 个函数中,GWO 的均值随着维度变大而显著增加,LMGWO 的均值保持不变。这表明维数的不断增加对 LMGWO 的寻优能力影响不大,与 GWO 相比寻优性能更加突出,进一步验证了本文所提算法的优越性。



(a) F_1变化曲线　(b) F_2变化曲线　(c) F_3变化曲线

(d) F_4变化曲线　(e) F_5变化曲线　(f) F_6变化曲线

(g) F_7变化曲线　(h) F_8变化曲线　(i) F_9变化曲线

图 3　基于函数维数变化曲线的函数优化

Fig. 3　Function optimization based on the curve of function dimension change

4 求解最优潮流(OPF)问题

为了验证 LMGWO 算法的有效性和可行性,在标准 IEEE-30 总线测试系统模型上对算法进行了测试。该系统包括 6 台发电机、4 台变压器、9 台分流器和 41 条支路。IEEE 30 母线系统单线如图 4 所示。图 4 中母线 1 为平衡母线,母线 2、5、8、11、13 为电压控制(Voltage Control)和无功功率(Reactive Power)母线,其余为有功功率(Active Power)和无功功率(Reactive Power)母线。本文假设变压器比及无功补偿输出为连续变量,最大迭代次数设置为 200 次,种群规模为 40,OPF 问题维度为 24。

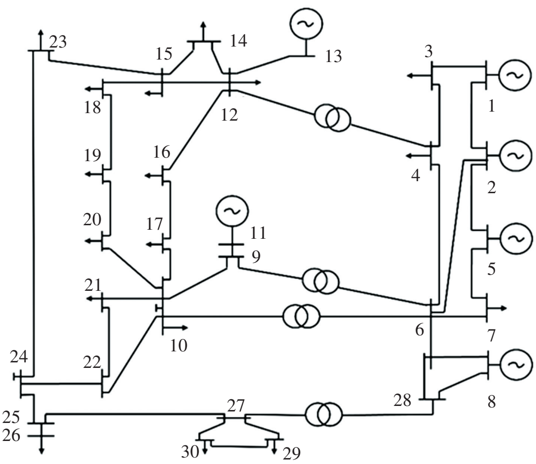


图 4 IEEE 30 总线测试系统单线图

Fig. 4 Single line diagram of IEEE 30 bus test system

4.1 案例 1:燃料成本(FC)最小化

最小化燃料成本是指通过各种手段和方法,将燃料成本控制在最低水平,以提高经济效益,同时也能够减少对环境的影响。将 LMGWO 算法与灰狼优化算法(GWO)[14]、算术优化算法(AOA)[18]、正弦余弦算法(SCA)[19]、猩猩优化算法(ChOA)[20]、鲸鱼优化算法(WOA)[21]算法进行对比实验,实验结果见表 5。由表 5 可知,优化后的 LMGWO 算法燃油成本为 799.394 4 \$/H。与初始情况相比,燃料成本降低了 11.37%,具有更加优越的性能。

表 5 不同算法在案例 1 上的比较结果

Table 5 Comparison results of different algorithms in Case 1

算法名称	燃油成本/(\$ · h^{-1})
GWO	799.962 4
AOA	799.921 7
SCA	801.970 0
ChOA	800.185 3
WOA	800.101 8
LMGWO	**799.394 4**

4.2 案例 2:有功功率损耗(APL)最小化

有功功率损耗(APL)是指电路中有功电流通过负载时所产生的功率损耗。有功功率损耗会导致电能转换效率降低,增加能源消耗和运营成本。因此,对于电力系统设计和运行来说,减小有功功率损耗是非常重要的。将 LMGWO 算法与灰狼优化算法(GWO)[14]、算术优化算法(AOA)[18]、正弦余弦算法(SCA)[19]、猩猩优化算法(ChOA)[20]、鲸鱼优化算法(WOA)[21]算法进行对比实验,实验结果见表 6。根据表 6 的实验结果,本文提出的 LMGWO 算法以有功功率损耗(APL)最小为目标,优于其他用于求解最优潮流(OPF)问题的对比算法。

表 6 不同算法在案例 2 上的比较结果

Table 6 Comparison results of different algorithms in Case 2

算法名称	有功功率损耗/MW
GWO	3.026 4
AOA	3.123 2
SCA	3.823 9
ChOA	3.160 0
WOA	3.516 5
LMGWO	2.969 1

5 结束语

本文提出了一种改进的灰狼优化算法(LMGWO),针对原始 GWO 算法在求解 OPF 问题时的性能进行了 2 方面的改进。将修正反向学习策略与透镜成像学习策略和乘除算子策略相结合,对 9 个具有不同特性的基准函数进行测试,并与现有元启发式算法进行对比实验。实验结果表明,LMGWO 比其他算法具有更好的稳定性和寻优性能。在实际应用案例中,将 LMGWO 算法和其他对比算法在 IEEE 30 节点标准测试系统模型上进行对比测试。实验结果表明,LMGWO 算法具有较好的性能。在未来的工作中,将使用 LMGWO 算法解决更困难的最优潮流(OPF)问题。

参考文献

[1] SALGADO R, BRAMELLER A, AITCHISON P. Optimal power flow solutions using the gradient projection method. Part 1: Theoretical basis[J]. IET Proceedings C (Generation, Transmission and Distribution), 1990, 137(6): 424-428.

[2] TINNEY W F, HART C E. Power flow solution by Newton's method[J]. IEEE Transactions on Power Apparatus and Systems, 1967 (11): 1449-1460.

[3] LEVI V A, NEDIC D P. Application of the optimal power flow model in power system education[J]. IEEE Transactions on Power

Systems, 2001, 16(4): 572-580.

[4] OLOFSSON M, ANDERSSON G, SÖDER L. Linear programming based optimal power flow using second order sensitivities[J]. IEEE Transactions on Power System, 1995, 10: 1691-1697.

[5] DING Xiaoying, WANG Xifan, SONG Yonghua, et al. The interior point branch and cut method for optimal power flow[C]// Proceedings of International Conference on Power System Technology. Kunming, China: IEEE, 2002, 1: 651-655.

[6] 刘自发,葛少云,余贻鑫. 基于混沌粒子群优化方法的电力系统无功最优潮流[J]. 电力系统自动化, 2005, 29(7):53-57.

[7] FARHAT M, KAMEL S, ATALLAH A M, et al. ESMA-OPF: Enhanced slime mould algorithm for solving optimal power flow problem[J]. Sustainability, 2022, 14(4): 2305.

[8] Attia A F, El Sehiemy R A, Hasanien H M. Optimal power flow solution in power systems using a novel Sine-Cosine algorithm [J]. International Journal of Electrical Power & Energy Systems, 2018, 99: 331-343.

[9] WARID W. Optimal power flow using the AMTPG-Jaya algorithm [J]. Applied Soft Computing, 2020, 91: 106252.

[10] WARID W, HIZAM H, MARIUN N, et al. Optimal power flow using the Jaya algorithm[J]. Energies, 2016, 9(9): 678.

[11] ABD E S, KAMEL S, EBEED M, et al. An improved version of salp swarm algorithm for solving optimal power flow problem[J]. Soft Computing, 2021, 25: 4027-4052.

[12] NGUYEN T T. A high performance social spider optimization algorithm for optimal power flow solution with single objective optimization[J]. Energy, 2019, 171: 218-240.

[13] ABDEL-RAHIM A M M, SHAABAN S A, RAGLEND I J. Optimal power flow using atom search optimization[C]//2019 Innovations in Power and Advanced Computing Technologies (i-PACT). Vellore, India :IEEE, 2019, 1: 1-4.

[14] MIRJALILI S, MIRJALILI S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.

[15] NUAEKAEW K, ARTRIT P, PHOLDEE N, et al. Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer [J]. Expert Systems with Applications, 2017, 87: 79-89.

[16] PRECUP R E, DAVID R C, PETRIU E M. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity [J]. IEEE Transactions on Industrial Electronics, 2017, 64(1): 527-534.

[17] SAXENA A, KUMAR R, DAS S. β-chaotic map enabled grey wolf optimizer[J]. Applied Soft Computing, 2019, 75: 84-105.

[18] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm [J]. Computer Methods in Applied Mechanics and Engineering, 2021, 376: 113609.

[19] MIRJALILI S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-based Systems, 2016, 96: 120-133.

[20] KHISHE M, MOSAVI M R. Chimp optimization algorithm[J]. Expert Systems with Applications, 2020, 149: 113338.

[21] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.