



一种离散鲸鱼算法及其应用

张 强, 郭玉洁*, 王 颖, 刘 馨

(东北石油大学计算机与信息技术学院 黑龙江 大庆 163318)

【摘要】针对鲸鱼优化算法 (WOA) 在求解高维复杂问题时存在收敛精度低, 缺少解决离散优化问题的能力等方面的不足, 提出了一种离散鲸鱼算法 (DWOA)。该算法引入收敛因子调控个体距离最优鲸鱼位置的远近程度, 利用惯性权重平衡算法的全局搜索和局部开发能力, 通过改进的 Sigmoid 函数对 WOA 进行离散化处理。9 个基准函数和油田措施规划方案的测试结果表明, DWOA 在收敛速度和寻优精度等方面均有较大的提升。

关 键 词 自适应惯性权重; 收敛因子; 离散鲸鱼算法; 油田措施规划; Sigmoid

中图分类号 TP18 **文献标志码** A **doi:**10.12178/1001-0548.2019116

A Discrete Whale Optimization Algorithm and Application

ZHANG Qiang, GUO Yu-jie*, WANG Ying, and LIU Xin

(School of Computer and Information Technology, Northeast Petroleum University Daqing Heilongjiang 163318)

Abstract A discrete whale optimization algorithm (DWOA) is proposed to overcome the defects of low convergence and the lack of ability to solve discrete optimization problems when solving high-dimensional complex problems. In DWOA, the convergence factor is introduced to adjust the distance of the individual from the optimal whale position, the adaptive inertia weight is designed to balance the global exploration and local exploitation ability, the whale optimization algorithm (WOA) is discretized by the improved Sigmoid function. The optimization experiments are conducted on the 9 benchmark functions and oilfield measures planning. Simulation results show that the proposed DWOA has a great improvement in convergence speed and convergence precision.

Key words adaptive inertia weight; convergence factor; discrete whale optimization algorithm (DWOA); oilfield measures planning; Sigmoid

鲸鱼优化算法 (WOA) 是文献 [1] 提出的一种新元启发式优化算法, 其原理是模仿座头鲸的狩猎策略, 将座头鲸的狩猎行为演化成随机优化的过程。目前, 许多研究学者将 WOA 算法应用到不同的工程领域, 文献 [2] 提出一种局部搜索策略的混合鲸鱼算法, 用于解决排列流水车间调度问题, 得到高效率的调度策略。文献 [3] 提出了多目标鲸鱼算法 (MOWOA), 用来优化 Elman 中性网络的权重和阈值, 可以更好地预测风速对电网的影响。文献 [4] 引入惯性权重策略来改进鲸鱼优化算法, 在求解水资源分配模型时取得了很好的优化结果。文献 [5] 将鲸鱼算法和人工神经网络结合起来, 用于求解 Ad-hoc 网络中的最优路径问题。文献 [6] 提出了一种随机的鲸鱼算法 (SWO) 来解决发电机组

调度问题, 实验证明 SWO 算法可以得到更低的燃料成本。文献 [7] 将鲸鱼优化算法应用于识别垃圾邮件发送者的社交网络。文献 [8] 提出了一种混合鲸鱼粒子群优化 (HWPSO) 算法优化电力系统, 实验表明 HWPSO 算法的性能最佳。文献 [9] 将鲸鱼算法应用到最优无功功率调度问题上, 通过实验与粒子群算法进行比较, 证明了鲸鱼算法在解决 ORPD 问题方面的稳健性和有效性。然而在求解高维多目标复杂工程问题时, WOA 算法也存在收敛精度不高, 易陷入局部最优解等问题。

为了更好地改善 WOA 的优化性能, 本文提出了一种离散鲸鱼优化算法 (DWOA), 在 DWOA 算法中引入收敛因子调节个体距离最优鲸鱼位置的远近程度, 在最优位置附近进行更加精细的局部搜索,

收稿日期: 2019-05-10; 修回日期: 2020-05-25

基金项目: 国家自然科学基金 (61702093); 黑龙江省自然科学基金 (F2018003)

作者简介: 张强 (1982-), 男, 博士, 教授, 主要从事智能进化算法、神经网络方面的研究。

通信作者: 郭玉洁, E-mail: gyujie@126.com

提高算法的局部寻优能力；此外，在鲸鱼个体位置更新公式中引入自适应惯性权值来协调算法的全局探索和局部开发能力，进一步改善算法的优化性能；最后利用改进的 Sigmoid 函数对 WOA 进行离散化处理保证了个体的多样性。对改进效果进行仿真验证，并将 DWOA 与其他 8 种优化算法在求解油田措施规划模型上进行对比，DWOA 取得了很好的寻优效果。

1 离散鲸鱼优化算法原理

1.1 基本鲸鱼优化算法

鲸鱼优化算法 (WOA) 是模拟座头鲸的狩猎行为而构建的数学模型，寻优过程主要由包围猎物、泡泡网捕食和随机捕食 3 部分组成。

1) 包围猎物

座头鲸在捕食过程中，可以识别猎物的位置并包围它们。WOA 算法模拟该行为时，以当前最优个体为参照，其他鲸鱼个体将它们的位置向最优个体位置的方向更新，此行为表示为：

$$D = |CX^*(t) - X(t)| \quad (1)$$

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

式中， t 代表当前迭代次数； A 和 C 是系数向量； $X^*(t)$ 代表最优解的位置向量； $X(t)$ 是位置向量； D 表示两者之间的距离。如果有更好的解决方案，则应在每次迭代中更新 $X^*(t)$ 。

向量 A 和 C 的公式为：

$$A = 2\alpha\gamma - \alpha \quad (3)$$

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

式中， α 在迭代过程中从 2~0 线性递减； γ 为 [0,1] 随机数组成的向量。

2) 泡泡网捕食

座头鲸在围攻猎物时，收缩包围的同时也在沿螺旋形路径运动。在优化过程中，假设收缩包围机制和螺旋位置机制的更新概率相同，都设置为 0.5。用数学模型模拟座头鲸的泡泡网捕食行为的收缩包围机制和螺旋更新位置机制，有：

$$X(t+1) = \begin{cases} X^*(t) - AD & p < 0.5 \\ D'e^{bl} \cos(2\pi l) + X^*(t) & p \geq 0.5 \end{cases} \quad (5)$$

式中， $D' = |X^*(t) - X(t)|$ ； p 为 [0, 1] 随机概率； l 为 [-1, 1] 随机概率； b 为定义螺旋形状的常数。

3) 随机捕食

随机捕食与泡泡网捕食不同之处在于该阶段不

是根据当前获得的最优个体，而是根据随机选择的鲸鱼个体作为参照鲸鱼来更新其他鲸鱼个体的位置，随机捕食的数学模型如下：

$$D = |CX_{\text{rand}}(t) - X(t)| \quad (6)$$

$$X(t+1) = X_{\text{rand}}(t) - AD \quad (7)$$

式中， X_{rand} 是一个随机位置向量。随机捕食设定了当系数变量 $|A| \geq 1$ ，随机选择鲸鱼个体作为参照鲸鱼，以此来寻找其他更合适的猎物以增强算法的搜索捕食能力，使该算法能在全局范围内进行搜索。

1.2 离散鲸鱼优化算法原理

1) 进化方式改进原理

标准的 WOA 算法在进行全局搜索的过程中，鲸鱼的位置是随机更新的，不能有效地平衡全局搜索和局部开发。为了提高鲸鱼算法的寻优能力，本文引入了自适应惯性权值 w 来调节算法的全局搜索能力，有助于算法跳出局部最优。在迭代初期，应给予较大的惯性权值，保证算法的全局搜索能力，而在迭代后期，为了保证算法的局部搜索能力，应给予较小的惯性权值，进而平衡 WOA 算法在优化过程中的全局探索和局部开发能力。利用 w 的动态线性变化来控制鲸鱼位置对新鲸鱼位置的影响，改进后的更新公式为：

$$w = w_{\min} + w_{\max} \cos(t/T_{\max}) \quad (8)$$

$$X(t+1) = w(X_{\text{rand}}(t) - AD) \quad (9)$$

式中， w 是惯性权值； w_{\min} 是惯性权值最小值； t 是当前迭代数； T_{\max} 为最大迭代次数。

进化理论研究表明，优秀的解周围很大程度上存在更多潜在优秀的解。标准 WOA 算法在迭代中进行局部搜索时，所有鲸鱼会逐渐向最优鲸鱼的位置移动，因此很可能使得所有鲸鱼位于局部最优位置附近，算法只能在局部最优解附近搜索，不能很好寻找到更优的解。为了提高 WOA 算法的局部搜索能力，本文提出一种非线性收敛因子，通过收敛因子 V 可以更好地调控最优解位置与其他个体之间的距离，发掘更多的潜在优秀解。使算法在进行局部开发时，可以在最优值附近更加全面精细去搜索优秀解，改进之后的位置矢量为：

$$V = \text{rand}(v_{\min} - v_{\max}) + v_{\min} \quad (10)$$

$$D = |(CX^*(t) - VX(t))| \quad (11)$$

$$X(t+1) = \begin{cases} X^*(t) - AD & p < 0.5 \\ D'e^{bl} \cos(2\pi l) + X^*(t) & p \geq 0.5 \end{cases} \quad (12)$$

式中, v_{\min} 表示收敛因子的最小值; v_{\max} 表示收敛因子的最大值; rand 表示 0~1 之间的随机数; 向量 \mathbf{D} 表示个体距离最优解位置之间的远近程度。

2) 鲸鱼算法离散化原理

鲸鱼算法优化空间的值都是连续变量, 但在实际工程应用中所计算的变量都是离散的, 因此要对连续变量进行离散化处理。本文提出一种离散二进制鲸鱼算法编码方式, 利用映射函数完成对鲸鱼算法的离散化。在迭代过程中, 为了保证个体的位置只能为 0 或 1, 通常使用 Sigmoid 函数^[10-11]将位置压缩到 [0,1] 区间来表示个体取 1 的概率, 其公式为:

$$s(\mathbf{x}) = 1 / (1 + \exp(-\mathbf{x})) \quad (13)$$

$$\mathbf{x} = \begin{cases} 1 & \text{rand} < s(\mathbf{x}) \\ 0 & \text{其他} \end{cases} \quad (14)$$

Sigmoid 函数的核心是将连续搜索空间映射到离散搜索空间, 利用标准 Sigmoid 函数对鲸鱼个体位置映射的结果如图 1 所示。从图 1 可以看出, 映射结果大都集中在 0.5~0.76 之间, 若用式 (14) 对映射结果离散化处理, 个体容易产生“靠拢”现象, 会使算法过早收敛, 陷入局部最优。

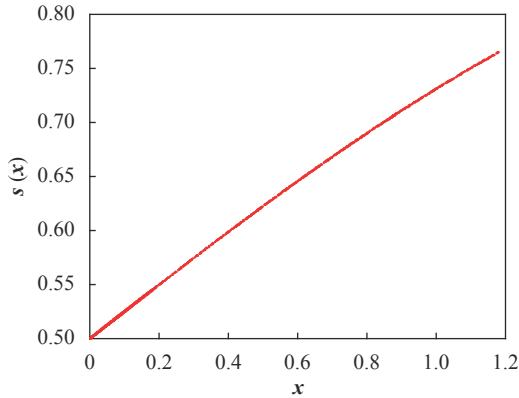


图 1 标准 Sigmoid 函数

为了提高算法的搜索能力, 本文通过对最优位置和个体位置的差值进行映射改进 Sigmoid 函数的映射公式, 映射结果如图 2 所示。

可以看出, 映射结果值分布在 0~0.9 之间, 更好地保证了离散化后个体的多样性, 为算法的优化过程奠定了多样性基础。改进后的 Sigmoid 函数为:

$$\mathbf{x} = 2\text{rand}(\text{VarSize}) \cdot (\mathbf{x}^* - \mathbf{x}) \quad (15)$$

$$s(\mathbf{x}) = 1 / (1 + \exp(-\mathbf{x})) \quad (16)$$

$$\mathbf{x} = \begin{cases} 1 & \text{rand} < s(\mathbf{x}) \\ 0 & \text{其他} \end{cases} \quad (17)$$

式中, \mathbf{x}^* 为当前最优个体; \mathbf{x} 为位置向量; VarSize 是维数大小。

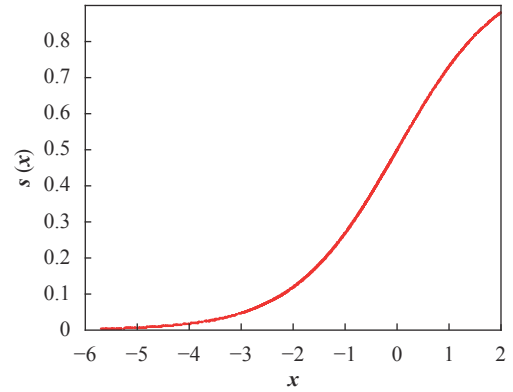


图 2 改进 Sigmoid 函数

1.3 算法步骤

DWOA 的具体操作步骤为:

1) 初始化鲸鱼种群 $\mathbf{x}_i (i = 1, 2, 3, \dots, Q)$, 计算每个个体 \mathbf{x}_i 的适应度值 F , 得到初始最优位置 \mathbf{x}^* 、最优解 F^* , 当前迭代次数 $t = 0$, 最大迭代次数为 T_{\max} ;

2) while $t < T_{\max}$

3) for $i = 1 : Q$

4) 应用式 (3)、式 (4) 更新参数

α, A, C, l 和 p ;

5) if $p < 0.5$

6) if $|A| < 1$

7) 应用式 (10)~(12) 更新策略得到新个体

$\mathbf{x}_i^{\text{new}}$;

8) else if $|A| \geq 1$

9) 应用式 (1)、式 (8) 和式 (9) 更新策略得到新个体 $\mathbf{x}_i^{\text{new}}$;

10) end if

11) else if $p \geq 0.5$

12) 应用式 (10)~(12) 更新策略得到新个体 $\mathbf{x}_i^{\text{new}}$;

13) end if

14) 应用式 (15)、式 (16) 对更新后的个体 $\mathbf{x}_i^{\text{new}}$ 进行离散化处理, 将其值映射到 [0,1] 区间;

15) if $\text{rand} < s(\mathbf{x}_i^{\text{new}})$

16) $\mathbf{x}_i^{\text{new}} = 1$;

17) else

18) $\mathbf{x}_i^{\text{new}} = 0$;

19) end if

```

20) end for
21) 计算每个新鲸鱼个体  $x_i^{\text{new}}$  的适应度值  $F^{\text{new}}$ ;
22) if  $F^{\text{new}}$  better than  $F^*$ 
23)    $x^* = x_i^{\text{new}}$ ;
24)    $F^* = F^{\text{new}}$ ;
25) end if
26) end while

```

2 实验仿真测试

本文将离散鲸鱼算法 (DWOA) 同标准鲸鱼算

法 (WOA)^[1]、灰狼算法 (GWO)^[13]、飞蛾扑火算法 (MFO)^[14]、布谷鸟算法 (CSA)^[15]、蝙蝠算法 (BAT)^[16]、多元宇宙算法 (MVO)^[17]、粒子群算法 (PSO)^[18]、乌贼算法 (CFA)^[19] 8 种优化算法进行实验对比, 参数设置如表 1 所示, 选用了 9 个 ($F1 \sim F9$) 常用的基准测试函数^[12] 如表 2 所示。对于每种测试函数, 算法的参数设置如下: 最大迭代次数为 100, 种群数为 40, 重复计算 10 次, 采用平均值 (Mean), 标准差值 (Std) 以及迭代中的最优值 (Best) 来评价算法的性能。 $F1 \sim F9$ 的迭代图分别如图 3~图 11 所示。

表 1 算法参数设置

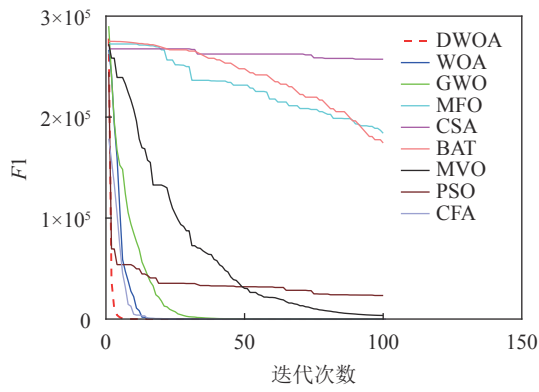
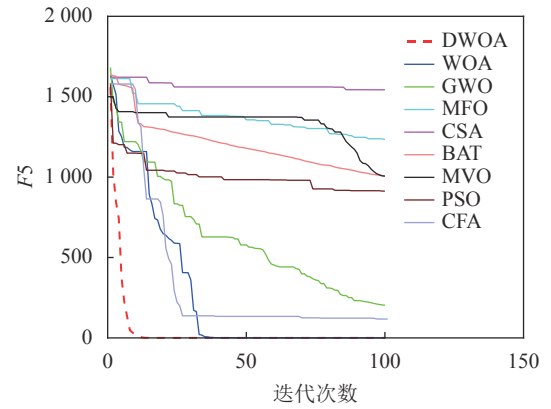
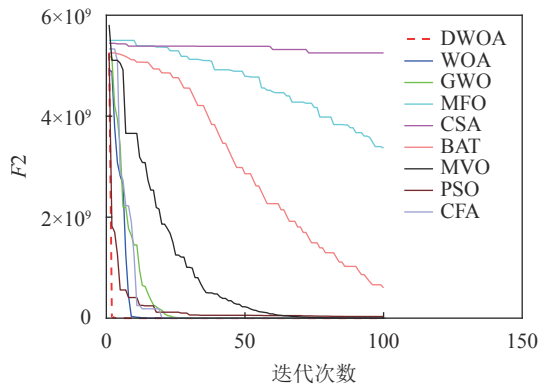
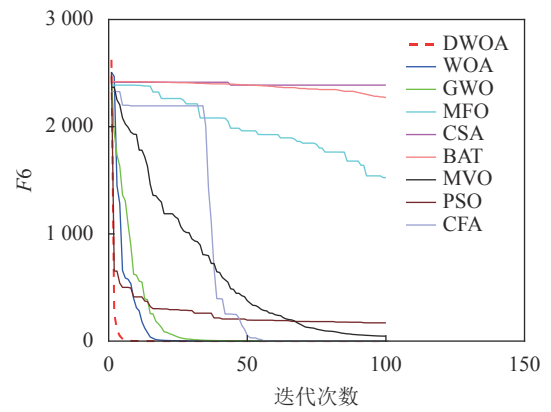
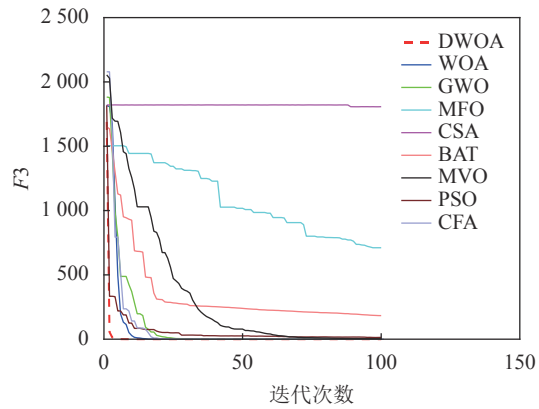
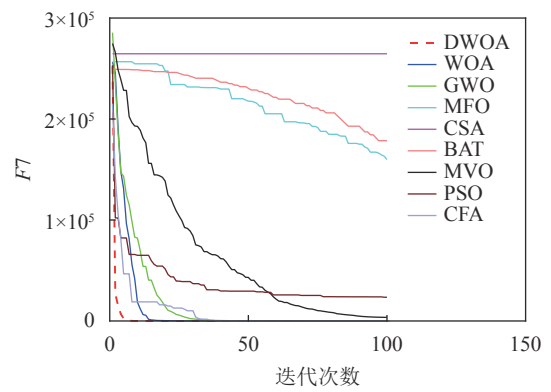
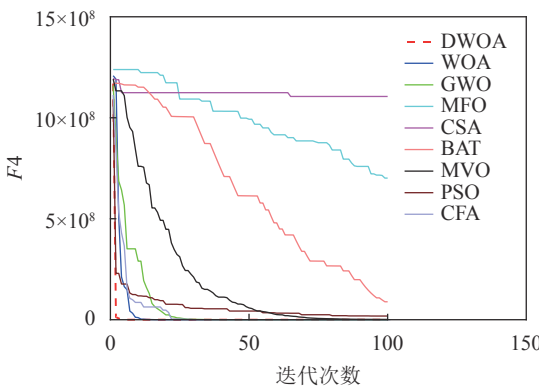
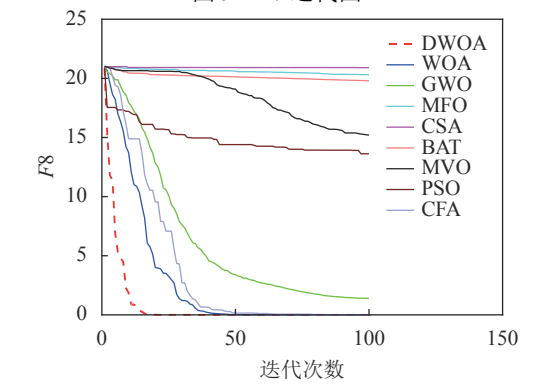
算法	参数设置
DWOA	$v_{\min} = 0, v_{\max} = 1.5, b = 1, w_{\min} = 0.1$
WOA	定义螺旋形状的常数 $b = 1$
GWO	惯性因子 $w = 0.5$, 惯性递减率 $w_{\text{damp}} = 0.99$, 加速因子: $c1 = 1, c2 = 2$
MFO	定义螺旋形状的常数 $b = 1$
CSA	布谷鸟蛋发现概率 $pa = 0.25$, 步长控制量 $\alpha = 1$
BAT	响度 $A = 0.5$, 脉冲发射率 $r = 0.5$, 最小频率 $Q_{\min} = 0$, 最大频率 $Q_{\max} = 2$
MVO	虫洞存在率最大值 $WEP_{\max} = 1$ 和最小值 $WEP_{\min} = 0.2$, 开发准确率 $p = 0.6$
PSO	惯性因子 $w = 0.5$, 惯性递减率 $w_{\text{damp}} = 0.99$, 加速因子: $c1 = 1, c2 = 2$
CFA	伸展度常数: $r1 = 2, r2 = -1$, 可见度常数: $v1 = -1.5, v2 = 1.5$

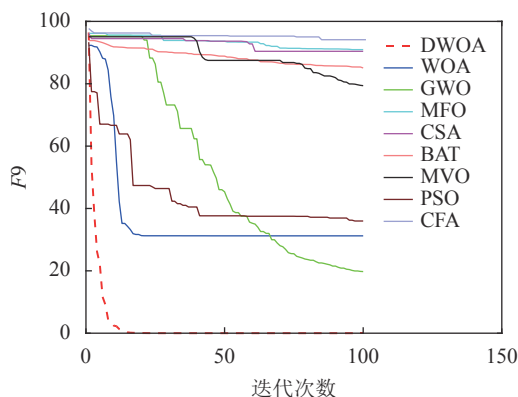
表 2 基准测试函数

名称	测试函数	公式	初始范围	维数	最优值
$F1$	Sphere	$\min \sum_{i=1}^D x_i^2$	$[-100, 100]$	100	0
$F2$	Penalized	$\min 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} \frac{(x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})]}{(x_D - 1)^2 [1 + \sin^2(2\pi x_D)]} \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	$[-50, 50]$	100	0
$F3$	Quartic	$\min \sum_{i=1}^D i x_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	100	0
$F4$	Rosenbrock	$\min \sum_{i=1}^D [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-30, 30]$	100	0
$F5$	Rastrigrin	$\min \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	100	0
$F6$	Griewank	$\min \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	100	0
$F7$	Step	$\min \sum_{i=1}^D (x_i + 0.5)^2$	$[-100, 100]$	100	0
$F8$	Ackley	$\min -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + \exp(1)$	$[-32, 32]$	100	0
$F9$	Schwefel 2.21	$\min \{ x_i , 1 \leq i \leq D\}$	$[-10, 10]$	100	0

表 3 为 9 种算法在 9 个测试函数中的实验结果对比, 从表 3 可以看出, 在函数条件设定相同的情况下, DWOA 算法能够找到 $F5$ 和 $F6$ 的理论最优值, 对于 $F1$ 、 $F3$ 、 $F8$ 、 $F9$ 函数来说, 虽

然 DWOA 精度没有达到理论最优, 但算法的搜索结果无限接近理论最优值, 与其他 8 个算法的实验结果相比, 在寻优精度上有了很大的提高, 也更加稳定。

图3 $F1$ 迭代图图7 $F5$ 迭代图图4 $F2$ 迭代图图8 $F6$ 迭代图图5 $F3$ 迭代图图9 $F7$ 迭代图图6 $F4$ 迭代图图10 $F8$ 迭代图

图 11 F_9 迭代图

以最优值和平均值为评判标准, 通过比较 9 种算法的优化结果可知, 在每个测试函数都是 100 维的条件下, DWOA 寻优结果最好, 分析其原因在

于: 在进化中引入自适应惯性权值可以更好地平衡算法的全局搜索和局部开发能力, 克服了算法在求解某些高维复杂函数问题时收敛速度慢, 容易陷入局部最优的缺点, 从 $F_1 \sim F_9$ 的运行结果可以看出, DWOA 的算法的收敛速度更快。

以方差为评判标准, DWOA 的收敛精度和方差都优于其他 8 种算法, 分析主要原因是: 在迭代过程中其他改进算法的个体会逐渐集中在最优个体的位置附近, 无法跳出局部最优, 而本文收敛因子可以调节个体距离最优位置的远近程度, 增加了算法局部搜索的精度, 有利于进行局部精细挖掘。从 $F_1 \sim F_9$ 的运行结果可以看出, DWOA 算法寻优结果更好, 证明其跳出局部最优能力得到了提升。

表 3 实验结果对比表

名称	指标	DWOA	WOA	GWO	MFO	CSA	BAT	MVO	PSO	CFA
F_1	Mean	3.20×10^3	1.18×10^4	2.17×10^4	2.16×10^5	2.74×10^5	2.15×10^5	7.18×10^4	3.63×10^4	6.06×10^4
	Std	2.65×10^4	4.70×10^4	5.37×10^4	2.54×10^4	3.63×10^3	2.51×10^4	7.26×10^4	2.70×10^4	1.10×10^5
	Best	1.58×10^{-31}	1.95×10^{-15}	$20.854\ 37$	1.75×10^5	2.71×10^5	1.57×10^5	5.11×10^3	1.83×10^4	1.01×10^{-7}
F_2	Mean	4.79×10^7	2.03×10^8	3.51×10^8	2.89×10^9	4.66×10^9	2.75×10^9	1.09×10^9	1.36×10^8	4.78×10^9
	Std	4.63×10^8	8.40×10^8	1.05×10^9	5.40×10^8	6.71×10^{-6}	1.32×10^9	1.53×10^9	5.71×10^8	1.44×10^8
	Best	5.000 743	5.019 411	41.888 09	2.00×10^9	4.66×10^9	6.32×10^8	4.45×10^6	1.14×10^7	4.63×10^9
F_3	Mean	17.392 93	1.01×10^2	94.979 35	1.31×10^3	1.66×10^3	5.43×10^2	3.03×10^2	1.09×10^2	1.14×10^3
	Std	1.69×10^2	3.61×10^2	2.83×10^2	3.36×10^2	64.008 79	1.84×10^2	4.73×10^2	2.17×10^2	7.80×10^2
	Best	0.001 618	0.002 505	0.065 976	7.24×10^2	1.62×10^3	3.52×10^2	3.926 793	25.844 82	2.983 437
F_4	Mean	1.07×10^7	4.78×10^7	7.72×10^7	8.98×10^8	1.08×10^9	5.82×10^8	2.11×10^8	4.24×10^7	1.05×10^9
	Std	1.04×10^8	1.97×10^8	2.03×10^8	1.68×10^8	4.39×10^7	3.99×10^8	3.01×10^8	1.2×10^8	5.26×10^7
	Best	98.215 81	98.224 25	1.16×10^3	5.79×10^8	1.01×10^9	6.86×10^7	2.11×10^6	8.28×10^6	1.02×10^9
F_5	Mean	47.658 84	1.60×10^2	6.72×10^2	1.27×10^3	1.55×10^3	1.11×10^3	1.25×10^3	9.68×10^2	2.24×10^2
	Std	2.15×10^2	4.05×10^2	3.61×10^2	1.53×10^2	30.454 32	1.49×10^2	1.35×10^2	1.14×10^2	2.73×10^2
	Best	0	1.14×10^{-13}	2.30×10^2	1.05×10^3	1.52×10^3	9.12×10^2	9.16×10^2	8.93×10^2	75.222×10^8
F_6	Mean	30.051 38	1.00×10^2	1.83×10^2	2.07×10^3	2.46×10^3	2.16×10^3	6.13×10^2	4.22×10^2	1.43×10^2
	Std	2.44×10^2	3.97×10^2	4.54×10^2	2.48×10^2	37.983 83	40.477 91	6.38×10^2	2.44×10^2	4.79×10^2
	Best	0	9.99×10^{-16}	1.069 829	1.60×10^3	2.43×10^3	2.07×10^3	50.082 25	2.60×10^2	0.288 459
F_7	Mean	3.48×10^3	1.25×10^4	2.03×10^4	2.28×10^5	2.64×10^5	2.53×10^5	6.97×10^4	3.99×10^4	2.65×10^4
	Std	2.83×10^4	4.45×10^4	5.03×10^4	3.03×10^4	2.93×10^{-10}	2.63×10^4	7.11×10^4	3.07×10^4	7.47×10^4
	Best	5.145 72	5.960 822	30.370 96	1.64×10^5	2.64×10^5	1.95×10^5	4.56×10^3	2.15×10^4	23.795 77
F_8	Mean	0.757 705	3.507 511	6.014 612	20.639 71	20.810 35	20.290 52	20.381 43	15.839 76	20.360 69
	Std	3.010 918	6.414 977	6.438 818	0.169 238	2.86×10^{-14}	0.260 663	0.166 892	1.010 50	0.185 286
	Best	4.44×10^{-15}	4.87×10^{-8}	0.865 164	20.324 10	20.810 35	19.945 66	20.145 31	14.566 14	20.219 25
F_9	Mean	2.742 325	37.247 32	53.052 34	93.361 84	92.654 89	88.703 08	89.242 66	44.175 33	95.352 73
	Std	12.043 100	16.869 06	28.906 30	1.776 699	1.891 686	2.489 666	5.257 728	12.001 02	0.728 189
	Best	4.58×10^{-9}	31.222 24	19.763 52	90.948 71	90.402 55	85.064 51	79.386 84	35.983 38	94.125 31

综上所述, 改进的策略提高了 DWOA 算法寻优性能, 其优化性能比其他算法更加有效, 能够有效解决高维复杂数值优化问题。

3 基于 DWOA 的油田措施方案优化

本文利用 DWOA 优化油田措施规划模型, 以

年度利润投入比和累积利润投入比最大为目标函数, 建立油田措施规划模型, 通过和其他 8 种算法优化结果进行对比, 分析 DWOA 在优化油田措施规划中的性能。

3.1 油田措施规划数学模型

1) 目标函数

油田措施规划问题是在满足油田开采要求的各种约束条件的前提下, 合理分配开采井组, 使油田的年度利润投入比和累积利润投入比最大, 其目标函数为:

$$\max F_1 = \frac{\sum_{i=1}^N \sum_{j=1}^K (x1_{ij} + y1_{ij}) \cdot a - \sum_{i=1}^N \sum_{j=1}^K z1_{ij} \cdot b - \sum_{i=1}^N \sum_{j=1}^K w1_{ij} \cdot c}{\sum_{i=1}^N \sum_{j=1}^K x1_{ij} \cdot \text{price}_{ij} + y1_{ij} \cdot \text{price}_{ij}} \quad (18)$$

$$\max F_2 = \frac{\sum_{i=1}^N \sum_{j=1}^K (x2_{ij} + y2_{ij}) \cdot a - \sum_{i=1}^N \sum_{j=1}^K z2_{ij} \cdot b - \sum_{i=1}^N \sum_{j=1}^K w2_{ij} \cdot c}{\sum_{i=1}^N \sum_{j=1}^K x2_{ij} \cdot \text{price}_{ij} + y2_{ij} \cdot \text{price}_{ij}} \quad (19)$$

式中, F_1 是年度利润投入比; F_2 是累积利润投入比; N 为井数目; K 为措施数; a 、 b 、 c 分别是吨油成本、吨水成本、吨液成本; $x1_{ij}$ 为第*i*口油井第*j*个措施的年度增油量; $y1_{ij}$ 为第*i*口水井第*j*个措施的年度增油量; $z1_{ij}$ 为第*i*口井第*j*个措施的年度增液量; $w1_{ij}$ 为第*i*口井第*j*个措施的年度增注量; $x2_{ij}$ 为第*i*口油井第*j*个措施的累积增油量; $y2_{ij}$ 为第*i*口水井第*j*个措施的累积增油量; $z2_{ij}$ 为第*i*口井第*j*个措施的累积增液量; $w2_{ij}$ 为第*i*口井第*j*个措施的累积增注量; price_{ij} 是第*i*口井措施数为*j*的价格。

2) 约束条件

① 增油量约束为:

$$\sum_{i=1}^N \sum_{j=1}^K x1_{ij} = t_o \quad (20)$$

$$0 \leq \sum_{i=1}^N \sum_{j=1}^K y1_{ij} - t_e \leq t_e \delta \quad (21)$$

式中, t_o 是油井增油目标; t_e 是水井增油目标; $\delta=0.03$ 为误差百分比。

② 增液量和增注量约束为:

$$\left| t_l - \sum_{i=1}^N \sum_{j=1}^K z1_{ij} \right| \leq t_l \delta \quad (22)$$

$$\left| t_c - \sum_{i=1}^N \sum_{j=1}^K w1_{ij} \right| \leq t_c \delta \quad (23)$$

式中, t_l 是增液量上限; t_c 是增注量上限。

③ 措施数上限约束为:

$$0 \leq \sum_{i=1}^N \sum_{j=1}^K K_{ij} \leq \text{Limit}_{ij} \quad (24)$$

式中, Limit_{ij} 是采油井的措施数上限。

3.2 实验分析

结合采油厂某区块当年的生产任务, 采取的措施和工作目标包括: 油井压裂为 61 口、油井补孔为 161 口、油井堵水为 47 口、油井换泵为 109 口、水井压裂为 50 口、水井酸化为 705 口、水井补孔为 36 口、水井测调为 404 口、水井细分为 293 口、水井周期为 151 口、水井浅调为 434 口、水井深调为 39 口, 共计 2 490 口措施井。其中, 油井增油目标为 7.1×10^4 T, 水井增油目标为 7.5×10^4 T, 年增液量上限为 18×10^4 T, 年增注量上限为 30×10^4 m³, 误差百分比 δ 为 0.03。

表 4 F_1 算法结果对比

算法	年度利润投入比(F_1)			
	最优值	最差值	平均值	方差
DWOA	3.264 96	2.908 77	3.154 61	0.133 31
WOA	3.068 76	2.789 15	3.021 25	0.090 70
GWO	3.048 40	2.793 57	3.014 22	0.069 26
MFO	2.845 29	2.793 66	2.838 76	0.012 63
CSA	2.963 69	2.774 62	2.925 23	0.057 86
BAT	3.029 23	2.774 62	2.994 19	0.072 93
PSO	3.206 33	2.791 10	3.058 15	0.178 57
MVO	2.923 92	2.863 85	2.922 72	0.008 41
CFA	3.236 10	2.754 66	3.097 33	0.133 67

表 5 F_2 算法结果对比

算法	累积利润投入比(F_2)			
	最优值	最差值	平均值	方差
DWOA	3.285 33	3.027 14	3.148 20	0.121 66
WOA	3.243 67	2.961 45	3.200 67	0.085 82
GWO	3.205 08	2.880 92	3.176 78	0.068 54
MFO	3.108 96	2.851 16	3.080 36	0.052 42
CSA	3.220 77	2.934 19	3.198 03	0.055 72
BAT	3.272 53	2.934 19	3.229 06	0.093 87
PSO	3.250 27	2.810 88	3.140 66	0.134 88
MVO	3.164 60	3.011 13	3.161 53	0.021 48
CFA	3.184 46	2.865 56	3.085 87	0.086 24

为验证本文算法求解油田措施规划模型的高效性, 采用 9 种优化算法针对 2 490 口措施井进行油水井措施方案优选, 并将实验结果进行对比分析。算法的参数设置如表 3 所示, 种群大小为 100, 最大迭代次数为 1 000, 每 20 代取一次平均值作为最

优值。

本文实验解决的主要问题是使年度利润投入比和累积利润投入比最大化,表 4、表 5 总结了 9 种算法优化得到的最优值、最差值、平均值、方差,图 12、图 13 表示的是 9 种优化算法运行时的结果相对于迭代次数的变化情况。

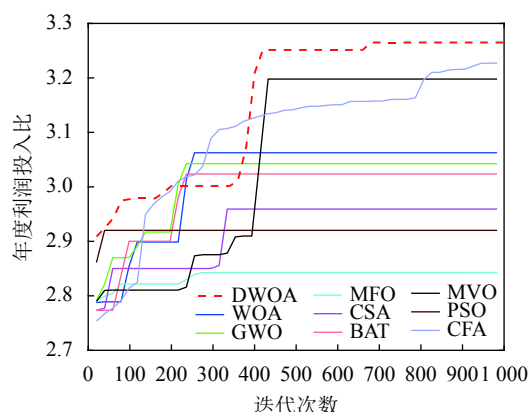


图 12 年度利润投入比对比图

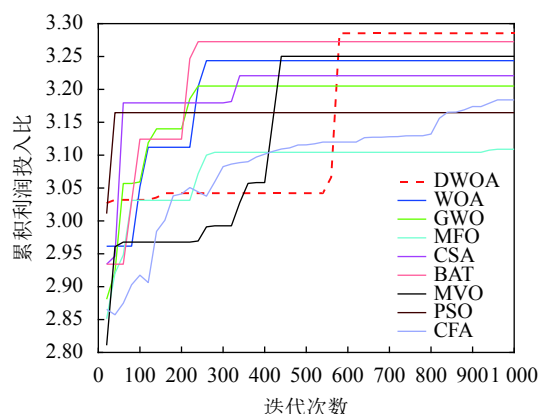


图 13 累积利润投入比对比图

由表 4 和表 5 分析可知,从最优值、最差值方面来看,在求解油田措施模型问题中,DWOA 得到的优化结果最好。在迭代过程中,DWOA 算法引入了收敛因子,可以更好地调控最优个体和其他个体之间的距离,精细算法的局部搜索能力,有更大的潜力去寻找潜在的优秀解;此外,随着迭代的进行,自适应权值逐渐减小,可以平衡迭代后期算法的全局搜索和局部开发能力,提高算法的收敛速度;在对位置更新方式进行离散化处理时,应用改进的 Sigmoid 函数离散化位置向量,可以更好地对 0-1 编码方式进行概率映射,保证了经过离散化处理后,粒子的多样性不会骤减,避免了算法过早陷入局部最优解,可以充分挖掘搜索空间。

从方差和平均值方面可以看出,DWOA 算法

相较于其他算法有较好的稳定性。从图 12 和图 13 中可知,PSO 算法最早陷入局部最优,过早收敛严重影响了 PSO 算法的性能,BAT、GWO、MFO、WOA 算法在 250 代左右最优值不再发生变化,而 MVO 算法在 450 代以后也陷入局部最优,但由于收敛因子提高了 DWOA 算法跳出局部最优的能力,因此 DWOA 算法的收敛曲线在 600 代时仍有上升的趋势,能够寻找到更优的解。

综上所述,从实际应用结果验证来看,DWOA 算法在求解油田措施规划模型时,有着良好的开发和勘探能力,可以得到较优的结果,提高油田开采的经济效益。

4 结束语

鲸鱼优化算法是模拟座头鲸狩猎机制而提出的一种新的群智能优化算法,本文提出一种离散鲸鱼算法(DWOA),利用收敛因子来控制个体到最佳解位置的距离,提高算法的局部搜索能力;引入自适应惯性权值来提高算法的全局搜索能力,提高了 DWOA 算法的收敛精度和收敛速度;并用改进的 Sigmoid 函数离散化位置更新方式,保证种群进化过程中的多样性。基准测试函数实验验证改进策略的有效性和可行性,并成功地将其应用到油田措施规划模型优化问题中。结果表明,同其他 8 种优化算法相比,DWOA 算法求解质量取得了很好的效果。今后的研究工作主要一方面进行更多的实验来验证并改进算法的性能,另一方面研究本文算法在其他工程应用领域的可行性,以扩大算法的应用范围。

参考文献

- [1] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [2] ABDEL-BASSET M, MANOGARAN G, EL-SHAHAT D, et al. A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem[J]. *Future Generation Computer Systems*, 2018, 85: 129-145.
- [3] WANG Jian-zhou, DU Pei, NIU Tong, et al. A novel hybrid system based on a new proposed algorithm multi-objective whale optimization algorithm for wind speed forecasting[J]. *Applied Energy*, 2017, 208: 344-360.
- [4] YAN Zhi-hong, SHA Jin-xia, LIN Bin, et al. An ameliorative whale optimization algorithm for multi-objective optimal allocation of water resources in Handan, China[J]. *Water*, 2018, 10(1): 87.
- [5] YADAV H, LITHORE U, AGRAWAL N, et al. An

- enhancement of whale optimization algorithm using ANN for routing optimization in Ad-hoc network[J]. *International Journal of Advanced Technology and Engineering Exploration*, 2017, 4(36): 161-167.
- [6] MOHAMED F A, ABDELNASSER M, MAHMOUD K, et al. Accurate economic dispatch solution using hybrid whale-wolf optimization method[C]//Middle East Power Conference. Cairo, Egypt: IEEE, 2018: 922-927.
- [7] AL Z, ALA M, FARIS H, et al. Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts[J]. *Knowledge-Based Systems*, 2018, 153: 91-104.
- [8] LASKAR N M, GUHA K, CHATTERJEE I, et al. HWPSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems[J]. *Applied Intelligence*, 2018, DOI: 10.1007/s10489-018-1247-6.
- [9] MEDANI K B O, SAYAH S, BEKRAR A. Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system[J]. *Electric Power Systems Research*, 2017, DOI: 10.1016/j.epsr.2017.09.001.
- [10] MIRJALILI S, LEWIS A. S-shaped versus v-shaped transfer functions for binary particle swarm optimization[J]. *Swarm and Evolutionary Computation*, 2013, 9(4): 1-14.
- [11] SAREMI S, MIRJALILI S, LEWIS A. How important is a transfer function in discrete heuristic algorithms[M]. Berlin: Springer-Verlag, 2014: 625-640.
- [12] 龚文引. 差分演化算法的改进及其在聚类分析中的应用研究[D]. 武汉: 中国地质大学, 2010.
- GONG wen-yin. Differential evolution algorithm and its application in clustering analysis[D]. Wuhan: China University of Geosciences, 2010.
- [13] KENNEDY J, EBERHART R. Particle swarm optimization[C]//International Conference on Neural Networks. Perth, Australia: IEEE, 1995: 1942-1948.
- [14] MIRJALILI S, MIRJALILI S M, HATAMLOU A. Multi-verse optimizer: A nature-inspired algorithm for global optimization[J]. *Neural Computing and Applications*, 2016, 27(2): 495-513.
- [15] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [16] MIRJALILI S. Moth-flame optimization algorithm[M]. Amsterdam, Holland: Elsevier Science Publishers, 2015: 228-249.
- [17] YANG X S, DEB S. Cuckoo Search via Lévy flights[C]//Nature & Biologically Inspired Computing. Coimbatore, India: IEEE, 2009: 210-214.
- [18] YANG X S. A new metaheuristic bat-inspired algorithm[J]. *Computer Knowledge & Technology*, 2010, 284: 65-74.
- [19] EESA A S, BRIFCANI A M A, ORMAN Z. A new tool for global optimization problems: cuttlefish algorithm[J]. *Electrical and Computer Engineering*, 2014, 8(9): 1235-1239.

编辑 漆蓉