

基于改进鲸鱼优化算法的非线性方程组求解

李霞 刘沫然

河北科技师范学院数学与信息科技学院

DOI: 10.12238/ems.v6i11.10005

[摘要] 针对鲸鱼优化算法 (WOA) 收敛速度慢、全局搜索能力弱的缺点, 首先对 WOA 中的参数进行改进, 然后利用樽海鞘群算法 (SSA) 收敛速度快的优点, 将其引入到 WOA 中, 得到改进的鲸鱼优化算法 (AWOA), 最后把 AWOA 应用到非线性方程组的求解中。结果显示, AWOA 不仅提高了搜索解的个数, 而且所得解平均值的误差和标准差低于 WOA。因此, AWOA 相较于 WOA, 全局搜索能力更强, 求解精确更高, 稳定性更佳。

[关键词] 改进的鲸鱼优化算法; 樽海鞘群算法; 非线性方程组求解

Ameliorated Whale Optimization Algorithm for Solving Nonlinear Equations

Xia Li Moran Liu

School of Mathematics and Information Science and Technology, Hebei Normal University of Science and Technology

[Abstract] In view of the shortcomings of whale optimization algorithm (WOA), such as slow convergence speed and weak global search ability, parameters of WOA were first modified, and then salp swarm algorithm (SSA) was introduced into WOA. An ameliorated whale optimization algorithm (AWOA) was obtained and was applied to solving nonlinear equations. The results show that AWOA not only increases the number of search solutions, but also has lower error and standard deviation than WOA. Therefore, compared with WOA, AWOA has stronger global search ability, higher solution accuracy and better stability.

[Keywords] Ameliorated whale optimization algorithm; Salp swarm algorithm; Solving nonlinear equations

1 引言

受大自然生物自然行为现象的启发, 人们提出了元启发式算法。比如遗传算法, 其灵感来源于自然选择和自然遗传学的进化过程; 模拟退火, 其灵感来源于冶金中的加热和控制冷却操作; 群体智能算法, 其灵感来自鸟群的社会行为、动物捕食狩猎、细菌生长和鱼群。其中, Drigio 等^[1]提出了蚁群算法、Eberhart 等^[2]提出了粒子群算法、Karaboga 等^[3]提出了蜂群算法等。

鲸鱼优化算法是 Mirjalili 等^[4]在 2016 年提出的一种基于群智能的元启发式算法。鲸鱼优化算法灵感来源于座头鲸的捕食行为, 通过模拟座头鲸的捕食实现其开发与探测, 主要包括包围猎物、气泡网攻击方法、搜索猎物三个阶段。鲸鱼优化算法有结构简单, 参数少, 概念易懂, 全局优化性能好的特点。但也存在着易陷入局部最优, 全局搜索能力和局部开发能力不均衡等问题^[5]。樽海鞘群算法是 2017 年由 Mirjalili 等^[6]提出的一种新型仿生智能群算法。樽海鞘群算法在收敛速度和寻优能力方面表现较好, 但某些情况下可能会陷入局部最优, 无法找到全局最优解。

为了解决鲸鱼优化算法收敛速度慢、全局搜索能力弱, 首先对其参数进行改进, 然后利用樽海鞘群算法收敛速度快

的优点, 将其引入到鲸鱼优化算法中, 得到改进的鲸鱼优化算法, 最后将其应用到非线性方程组求解中, 并与鲸鱼优化算法计算结果进行对比。

2 鲸鱼优化算法 (WOA)

WOA 模拟了鲸鱼的觅食行为, 包括三个阶段: 包围猎物、气泡网攻击、搜索猎物。

(1) 包围猎物

座头鲸在狩猎时通过识别猎物的位置并且将猎物包围起来进行捕食。由于在搜索空间中不知道最优位置, 所以 WOA 最初假定目标猎物的最佳位置靠近最优解, 鲸鱼根据目标猎物位置或接近目标猎物最近点进行位置更新, 尝试更新为最佳位置, 位置更新公式如下式所示:

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

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

其中, D 为个体到最优个体的距离, t 为迭代次数, A 、 C 为系数函数, $X^*(t)$ 为随机个体位置, $X(t+1)$ 为迭代 $t+1$ 次后个体所处位置。 A 、 C 由公式 (3) 和 (4) 更新:

$$A = 2a \cdot r_1 - a \quad (3)$$

$$C = 2 \cdot r_1 \quad (4)$$

r_1 和 r_2 为 $[0,1]$ 范围内的随机数; a 的值从 2 至 0 线性递减, 更新公式表示为:

$$a = 2 - 2t/T_{max} \quad (5)$$

其中 T_{max} 为最大迭代次数, 系数函数 A 的范围在 $[-a, a]$, $|A| \leq 1$ 时, 鲸鱼捕食行为总是在一个搜索包围圈里, 也就是用收缩包围机制包围猎物, 为局部搜索阶段, 按照(2)更新鲸鱼位置。

(2) 气泡网攻击

座头鲸在捕食时, 会以螺旋运动的方式逐渐缩小包围圈, 在该算法中, 鲸鱼个体通过公式(6)和(7)来更新位置:

$$D' = |X^*(t) - X(t)| \quad (6)$$

$$X(t+1) = D' \cdot e^{bl} \cos(2\pi l) + X^*(t) \quad (7)$$

其中, D' 为鲸鱼与猎物之间的距离, b 为一个常数, 其作用是保持鲸鱼作螺旋运动, 常设为 1, l 是属于 $[-1,1]$ 之间的随机数。需要注意的是, 鲸鱼优化算法在缩小包围圈的同时会围绕猎物沿螺旋轨迹运动, 因此鲸鱼有 50% 的可能包围猎物, 也有 50% 的可能选择通过气泡网攻击更新公式:

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), & p \geq 0.5 \end{cases} \quad (8)$$

其中, p 是 $[0,1]$ 之间的随机数。

(3) 搜索猎物

在鲸鱼优化算法中, 每个参与捕食的鲸鱼都是一个可行解, 在猎物搜寻阶段, 由于鲸鱼捕食具有随机性, 所以每个可行解通过相互之间的位置关系来更新位置。在 WOA 中, 当 $|A| \geq 1$ 时, WOA 将会随机选择鲸鱼位置进行更新, 下面是更新公式:

$$D = |C \cdot X_{rand} - X| \quad (9)$$

$$X(t+1) = X_{rand} - A \cdot D \quad (10)$$

其中 X_{rand} 是当前种群中随机一个鲸鱼的位置信息。WOA 通过设定 A 的范围来增强全局搜索能力, 避免陷入局部最优。

3 樽海鞘群算法 (SSA)

樽海鞘常以一条链的形式进行移动, 位于链条最前端的称作领导者, 剩余都称为追随者。整个捕食行为由领导者决定捕食方向, 追随者跟在领导者的后面。假设种群个数为 N , 搜索空间为 D 维, F 为食物的位置, 领导者根据下列公式更新位置:

$$x_i^j = \begin{cases} F_j + r_1[(ub_j - lb_j)r_2 + lb_j], & r_3 \geq 0.5 \\ F_j - r_1[(ub_j - lb_j)r_2 + lb_j], & r_3 < 0.5 \end{cases} \quad (11)$$

其中, F_j 是食物源的第 j 维位置; x_i^j 表示第一个樽海鞘的第 j 维位置, 也就是领导者; ub_j 和 lb_j 分别代表搜索空间第 j 维的最大值和最小值; r_2 和 r_3 为 $[0, 1]$ 随机数, r_1 由下式确定:

$$r_1 = 2e^{-(4t/T)^2} \quad (12)$$

其中, t 为当前迭代次数, T 为最大迭代次数。

追随者位置更新公式为:

$$x_i^j = (x_i^j + x_i^{j-1}) / 2, 2 \leq i \leq N \quad (13)$$

其中, x_i^j 表示第 i 个追随者的第 j 维位置。

4改进的鲸鱼优化算法 (AWOA)

(1) 调整 WOA 算法的参数

首先, 对 WOA 中参数 A 进行改进, 修改式(5)为:

$$a = 2 \sin((1 - t/T_{max}) \times \pi / 2) \quad (14)$$

利用式(14)改进 A 的值。

然后, 对 WOA 中参数 l 进行改进, l 不完全是随机数, 通过式(15)计算得出:

$$l = (a_2 - 1) \cdot a_3 + 1 \quad (15)$$

其中 a_2 是从 -1.2 至 -1.8 呈非线性递减, 公式为:

$$a_2 = -0.6 \times \sin(t/T_{max} \times \pi / 2) - 1.2 \quad (16)$$

式(15)中 a_3 为 $[0, 1]$ 之间的随机数。

(2) 结合 SSA 搜索机制

在迭代过程中, 若当前代理的 P 小于 0.5, 则使用 WOA, 否则使用 SSA。更新后的 WOA 基于 A 的值更新当前代理, 若 $|A| \leq 1$, 则使用式(2)更新代理位置, 否则, 使用公式(10)更新代理位置。若 P 大于等于 0.5, 则使用 SSA, SSA 根据当前代理是领导者还是追随者来更新代理位置。如果当前代理是领导者 ($i \leq N/2$), 则使用式(11)更新代理位置, 否则使用式(2)更新代理位置。

AWOA 结合了 WOA 和 SSA, 增大了算法的灵活性, 可以根据具体问题进行调整和改进, 该算法有可能更有效地找到较好的解。AWOA 运行步骤如下: 首先初始化两种算法的参数, 然后为给定问题创建初始随机总体解, 再评估每个代理的适应度, 确定最佳代理。继续根据参数 P 确定使用的算法并更新代理位置。计算所得总体的适应度并确定此迭代的最佳代理。最后, 算法检查整个迭代过程是否满足要求。若满足, 则算法返回找到最佳代理, 否则, 重复之前步骤。

5基于改进鲸鱼优化算法的非线性方程组求解

5.1 非线性问题描述

设非线性方程组

$$F(x) = [f_1(x), f_2(x), \dots, f_n(x)]^T = 0 \quad (17)$$

其中, $a_i \leq x_i \leq b_i, i = 1, \dots, n$ 。 a_i 和 b_i 为变量 X 的分量 x_i 的上下限。求解方程组(17)等价于求解下面一个极值优化问题:

$$Find: X = (x_1, x_2, \dots, x_n)^T$$

$$s. t. \min g(x) = \sum_{j=1}^n (f_j(x))^2$$

当 $g(x)$ 为最小值时, 所对应的 X 即为方程组的解。

5.2 实验数据及分析

本次实验采用三个非线性方程组,从未知数的不同的个数和非线性方程组不同的难易程度分别对WOA和AWOA两种算法进行实验,尽量保证实验数据的可靠性。对于两种算法,迭代次数均为1000次,运行次数为50次分别得出两种算法搜到解的次数、平均值误差以及标准差。

$$\text{例 1} \begin{cases} f_1(x) = x_1^2 - x_2 - 1 = 0, \\ f_1(x) = (x_1 - 2)^2 + (x_2 - 0.5)^2 - 1 = 0, \end{cases}$$

其中, $0 \leq x_1, x_2 \leq 2$, 精确解为 $x_1^* = (1.546342, 1.391174)^T$,

$$x_2^* = (1.067412, 0.139460)^T。$$

表1 两种算法搜到解的次数对比

算法	WOA	AWOA
$x_1^* = (1.546342, 1.391174)^T$	28	20
$x_2^* = (1.067412, 0.139460)^T$	22	30
总共搜到解次数	50	50

表2 两种算法误差分析

算法	平均值误差	标准差
WOA	3.11074160078571e-08	1.01880737456109e-07
AWOA	8.707457945122699e-15	9.008120935811087e-15

例1为代数方程组。表1是WOA和AWOA两种算法求出解的次数对比,看出两种算法都得出50组解。表2是两种算法的平均值误差和标准差对比,WOA得出的解的平均值误差约是AWOA得出的解的平均值误差的 10^7 倍,所以AWOA得出的近似解比WOA得出的近似解更为精确;从标准差来看,AWOA得出的结果相较于WOA更为稳定。

$$\text{例 2} \begin{cases} f_1(x) = (x_1 - 5x_2)^2 + 40 \sin^2(10x_3) = 0, \\ f_1(x) = (x_2 - 2x_3)^2 + 40 \sin^2(10x_1) = 0, \\ f_1(x) = (3x_1 + x_3)^2 + 40 \sin^2(10x_2) = 0, \end{cases}$$

其中, $-1 \leq x_1, x_2, x_3 \leq 1$, 精确解为 $x^* = (0, 0, 0)^T$ 。

表3 两种算法搜到解的次数对比

算法	WOA	AWOA
$x^* = (0, 0, 0)^T$	13	18
总共搜到解次数	13	18

表4 两种算法误差分析

算法	平均值误差	标准差
WOA	2.27957827114082e-01	7.35431413456074e-01
AWOA	4.120520331440589e-02	1.272355287781861e-01

例2为超越方程组。表3表明WOA搜索到解的次数为13次,AWOA搜索到解的次数为26次,看出AWOA算法搜索到解的成功率更高;表4表明WOA得出的解的平均值误差约是AWOA得出的解的平均值误差的5倍,所以AWOA得出的近似解比WOA得出的近似解更为精确;从标准差来看,AWOA得出的结果相较于WOA更为稳定。

$$\text{例 3} \begin{cases} f_1(x) = x_1^3 + x_1^2 x_2 x_3 + x_1 x_2 x_3^2 + 2x_2 - 3 = 0, \\ f_2(x) = x_1^5 x_2^4 + x_1 x_2 x_3^2 + x_2^2 x_3 + 3x_1 - 4 = 0, \\ f_3(x) = x_3^5 + x_1^2 x_2^2 + x_1 x_3 + 3x_1 - 4 = 0, \end{cases}$$

其中 $-1 \leq x_i \leq 1, i = 1, 2, 3$, 精确解为 $x^* = (1, 1, 0)^T$ 。

表5 两种算法搜到解的次数对比

算法	WOA	AWOA
$x^* = (1, 1, 0)^T$	49	50
总共搜到解次数	49	50

表6 两种算法误差分析

算法	平均值误差	标准差
WOA	7.53573060216427e-04	4.45599220575255e-03
AWOA	1.383844744821745e-17	4.333646256911374e-17

表5表明WOA搜索到解的次数为49次,AWOA搜索到解的次数为50次,看出本文算法搜索到解的成功率更高;表6表明的WOA平均值误差约是AWOA得出的解的平均值误差的 10^{14} 倍,所以AWOA得出的近似解比WOA得出的近似解更为精确;从标准差来看,AWOA得出的结果相较于WOA更为稳定。

6 结论

本文首先对WOA进行了改进和提高,得到了AWOA。然后利用AWOA对不同的非线性方程组求解,并对计算结果进行了对比分析。结果表明,AWOA提高了搜索到的非线性方程组解的个数、精度和稳定性。因此,相较于WOA,AWOA能更有效的解决非线性方程组的求解问题。

[参考文献]

[1]Dorigo M, Maniezzo V, Colomi A. Ant system: optimization by a colony of cooperating agents[J]. IE EE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 1966, 26 (1): 29-41.

[2]Kennedy J, Eberhart R. Particle swarm optimization[C]// IEEE International Conference on Neural Networks, 1995: 1942-1948.

[3]Karaboga D. An idea based on honey bee swarm for numerical optimization:technical report-TRO6[R]. 2005: 1-10.

[4]Mirjalili S, Lewis A. The Whale Optimization Algorithm[J].Advances in Engineering Software, 2016, 95: 51- 67.

[5]武泽权,牟永敏. 一种改进的鲸鱼优化算法[J]. 计算机应用研究, 2020, 37 (12): 3618- 3621.

[6]Mirjalili S, Gandomi A H, Mirjalili S Z, et al. Slap Swarm Algorithm: A bio-inspired optimizer for engineering design problems [J]. Advances in Engineering Software, 2017, 114: 163-191.

作者简介:李霞(1980-),女,副教授,博士,研究方向:基于数据驱动的智能建模与智能优化(通讯作者)。

基金项目:国家自然科学基金(62006069),秦皇岛市社科联研究课题(2024LX095),专利转化后续研发经费(3010102240127)