



计算机工程与应用
Computer Engineering and Applications
ISSN 1002-8331, CN 11-2127/TP

《计算机工程与应用》网络首发论文

题目: 柯西变异和自适应权重优化的蝴蝶算法
作者: 高文欣, 刘升, 肖子雅, 于建芳
网络首发日期: 2019-10-12
引用格式: 高文欣, 刘升, 肖子雅, 于建芳. 柯西变异和自适应权重优化的蝴蝶算法. 计算机工程与应用.
<http://kns.cnki.net/kcms/detail/11.2127.TP.20191012.0949.012.html>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

柯西变异和自适应权重优化的蝴蝶算法

高文欣, 刘升, 肖子雅, 于建芳

上海工程技术大学 管理学院, 上海 201620

摘要: 针对基本蝴蝶优化算法(Butterfly Optimization Algorithm, BOA)存在的收敛精度较低、容易陷入局部最优解的问题, 提出柯西变异和自适应权重优化的蝴蝶算法(Cauchy variation and adaptive weight butterfly optimization algorithm)。通过在全局位置更新处引入柯西分布函数进行变异, 在局部位置更新处引入自适应权重因子, 改进了蝴蝶算法的局部搜索能力; 并且引入动态切换概率 P 来权衡全局探索与局部开发过程的比重, 本文改进算法通过对多个单峰, 多峰和固定测试维度的函数进行求解, 结果表明, CWBOA 对大多数测试函数有更好的求解精度、速度和稳定性。

关键词: 蝴蝶优化算法; 自适应权重; 柯西变异; 动态切换概率; 高维

文献标志码: A 中图分类号: TP18 doi: 10.3778/j.issn.1002-8331.1907-0048

高文欣, 刘升, 肖子雅, 等. 柯西变异和自适应权重优化的蝴蝶算法. 计算机工程与应用

GAO Wenxin, LIU Sheng, XIAO Ziya, et al. Butterfly optimization algorithm based on cauchy variation and adaptive weight. Computer Engineering and Applications

Butterfly Optimization Algorithm Based on Cauchy Variation and Adaptive Weight

GAO Wenxin, LIU Sheng, XIAO Ziya, YU Jianfang

College of Management, Shanghai University of Engineering Sciences, Shanghai 201620, China

Abstract: Basic butterfly optimization algorithm (BOA), which has low convergence precision and easy to fall into the local optimal solution. Cauchy variation and adaptive weight butterfly optimization algorithm (CWBOA) is proposed. By introducing the cauchy distribution function at the global location update, the adaptive weighting factor is introduced at the local location update to improve the local search ability of the butterfly algorithm, and the dynamic handover probability is introduced to weigh the proportion of the local mining and global search process. The optimization precision is improved and the convergence effect of the algorithm is improved. Based on 14 basic test functions, the optimization experiment shows that the improved algorithm has higher precision, faster convergence and stronger algorithm stability.

Key words: butterfly optimization algorithm; adaptive weight; cauchy variation; dynamic switching probability; high-dimensional

基金项目: 国家自然科学基金(No.61075115, No.61673258); 上海市自然科学基金(No.19ZR1421600)。

作者简介: 高文欣(1995-), 女, 硕士研究生, 主要研究方向: 群智能计算、智能计算、大数据处理与分析; 刘升(1966-), 男, 通讯作者, 教授, 博士, 主要研究方向: 人工智能, 智能计算; 肖子雅(1994-), 女, 硕士研究生, 主要研究方向为智能算法、项目调度与优化; 于建芳(1992-), 女, 硕士研究生, 主要研究方向: 群智能算法、智能计算, 工程优化, 项目调度。

1 引言

群智能算法模拟自然界中动植物的行为,例如蚁群和蜂群,模仿他们的求偶行为,捕食行为,筑巢行为等。蚁群算法和粒子群算法是两种早期提出的算法,现如今又提出了许多新颖的算法比如象群游牧算法^[1],磷虾群算法^[2],蚁狮算法^[3],乌鸦搜索算法^[4],鸡群算法^[5]等。群智能算法也用于了一些问题的解决,例如,旅行商问题、无线传感器网络节点定位问题、支持向量机优化问题、投资组合优化问题等。

Sankalap Arora 和 Satvir Singh 两位学者,通过观察蝴蝶觅食行为而受到启发,提出了一种新的群智能优化算法——蝴蝶优化算法(Butterfly Optimization Algorithm)^[6]。该算法的主要思想是模拟蝴蝶的觅食和求偶行为实现对目标问题的求解。蝴蝶的行为可以描述为它们向食物源位置的合作运动。蝴蝶接收、感知和分析空气中的气味以确定食物源或配对伴侣的潜在方向。BOA 算法模拟此行为在搜索空间中寻找最优解。与现有的一些元启发式算法相比,基本 BOA 操作简单、调整的参数少、鲁棒性好,并在工程实践的初步应用中就取得了良好的效果。然而在求解较高维函数的问题时,BOA 具有与其它群智能仿生算法类似的问题,即容易陷入局部最优值、收敛性差等缺陷。为了改进蝴蝶算法容易陷入局部最优和收敛性能差等问题,文献[7]提出了基于莱维飞行的蝴蝶优化算法,在全局位置更新和局部位置更新处引入莱维飞行策略,提高了算法的搜索能力;文献[8]提出了一种基于学习自动机机制的蝴蝶优化算法,引入学习自动机机制,加速了全局搜索的速度,达到真正的全局最优。文献[9]提出了一种基于感觉模态变化的蝴蝶优化算法,采用动态变化的感觉模态参数策略,改变了收敛精度,提高了收敛速度;文献[10]提出了一种混合人工蜂群算法的蝴蝶优化算法,克服了蝴蝶优化算法

后期搜索能力不足,提高了算法的收敛速度。对于现存的 BOA 改进策略的研究,虽然在一定程度上改进了算法的寻优性能,但是一般的也只是针对蝴蝶优化中某一个更新策略进行改进或者是改变初始化种群的方式,并没有有效的改进全局寻优和局部寻优的盲目性。

基本的蝴蝶优化算法存在的依赖初始种群,收敛精度低和易陷入局部最优等问题,针对这些问题本文提出了一种多策略改进的蝴蝶优化算法。利用柯西分布函数对蝴蝶的全局位置更新进行变异,提升算法的全局搜索能力,在蝴蝶算法的局部位置更新处引入自适应惯性权重因子,改进了算法的局部开采能力,并且使用动态切换概率来平衡局部搜索和全局搜索的比重,提高了寻优性能。通过 14 个基准测试函数测试,结果表明改进算法的具有更高的收敛精度和鲁棒性。

2 蝴蝶优化算法

在自然界之中,蝴蝶可以使用他们的各种感官如:嗅觉、视觉、味觉、触觉和听觉去寻找食物和求偶,这些感觉能够帮助他们迁徙、躲避狩猎者以及帮助它们找到合适的地方产卵。在所有的感觉中最重要的是嗅觉,嗅觉能够帮助蝴蝶寻找食物(花蜜),即使在很远的地方也不例外。为了能够找到食物,蝴蝶使用感觉受体用于嗅觉,这些受体分散在蝴蝶的身体部位,如触角、腿和手掌等。这些受体实际上是蝴蝶身体表面的神经细胞,被称为化学感受器。这些化学感受器引导蝴蝶找到最佳的交配伙伴,以便继续强大的遗传系统。雄性蝴蝶能够通过信息素来识别雌性,这是雌性蝴蝶发出的气味分泌物引起特异性反应。

蝴蝶会产生一些强度与其适应性相关的香味,即当蝴蝶从一个位置移动到另一个位置时,它的适应性会相应地变化,香味会在远处传播,其他蝴蝶个体能够感知它,这就是蝴蝶个体如何与其他蝴蝶

共享个体信息，形成一个群体的社会知识网络。当一只蝴蝶能够闻到来自其他的蝴蝶分泌的香味的时候，它将会朝着香味最浓的方向移动，该阶段在算法中被称为全局搜索。在另一种情况下，当蝴蝶不能从周围感知香味时它将随机移动，这一阶段是局部搜索阶段。

针对上述行为，提出如下假设：

1.所有的蝴蝶都应该散发一些香味，使蝴蝶能够相互吸引

2.每只蝴蝶都会随机移动或向发出最多香味的蝴蝶移动。

3.蝴蝶的刺激强度受目标函数的影响或决定。

4.全局搜索和局部搜索使用切换概率 p 来控制，受到物理接近度以及风、雨、雷、电等各种其他自然因素，局部搜索和全局搜索中的切换概率 p 具有重要意义。

在 BOA 算法中，每一只蝴蝶有它自己独特的感觉和个体感知能力。这同时也是区别于其他的元启发式算法的一个主要特征。蝴蝶个体产生的香味数学公式如下。

$$f = cI^a \quad (1)$$

其中 f 是香味的感知强度，即香味被其他蝴蝶感知的强度， c 是感觉模态， I 是刺激强度， a 是依赖于模态的幂指数，它解释了不同程度的吸收。本文在 $[0,1]$ 范围内取 a ，参数 a 是取决于模态的功率指数(在此为香味)，这意味着它表征吸收的变化。

在每次迭代中，搜索空间中的所有蝴蝶移动到新位置，然后评价它们的适应值。该算法首先计算解空间中不同位置上所有蝴蝶的适应度值。然后这些蝴蝶将通过计算公式(1)在它们的位置产生香味。在全局搜索阶段，蝴蝶朝着最优的蝴蝶(g^*)移动，它可以用公式(2)表示。

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (2)$$

其中 x_i^t 是第 t 次迭代中第 i 只蝴蝶的解向量 x_i ， g^* 表示在当前迭代的所有解中的最优解。第 i 只蝴蝶发出的香味量用 f_i 表示， r 是 $[0,1]$ 之间的随机数。在局部搜索，位置更新公式如下。

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \quad (3)$$

x_j^t 和 x_k^t 表示在第 t 次迭代解空间中的第 j 和第 k 只蝴蝶的解向量，若 x_j^t 和 x_k^t 属于同一个种群，并且 r 是 $[0,1]$ 之间的随机数，表示局部随机游走。基本蝴蝶算法的伪代码如表 1 所示。

表 1 基本蝴蝶算法的伪代码

```

基本蝴蝶算法的伪代码
输入：目标函数  $f(x)$ ，蝴蝶群体的规模  $N$ ，刺激浓度  $I$ ，感觉模态  $c$ ，幂指数  $a$  的初始值，全局更新和局部更新的转换概率  $p$ ，最大迭代次数  $MaxIter$ 
1.初始化种群
2.While  $t < MaxIter$ 
3.   for  $i=1: N$ 
4.     计算每只蝴蝶的香味浓度
5.   end for
6.   找到最优蝴蝶个体  $g^*$ 
7.   for  $i=1: N$ 
8.     随机产生  $[0,1]$  之间的随机数  $r$ 
9.     if  $rand > p$ 
10.      进行全局位置更新，按照公式(2)进行计算
11.     else
12.      进行局部随机游走，按照公式(3)进行计算
13.     end if
14.     对新得到的解  $x_i^{t+1}$  进行评估，如果  $x_i^{t+1}$  比  $x_i^t$  更好，就用  $x_i^{t+1}$  代替  $x_i^t$ 
15.   end for
16.   更新  $a$  的值
17.end while
输出：全局最优解

```

3 蝴蝶优化算法的改进

为了改进蝴蝶算法容易陷入局部最优和收敛精度低的问题，本文从三个方面对蝴蝶算法进行改进。首先通过引入柯西分布函数的方法对全局搜索的蝴蝶位置信息进行变异，提高蝴蝶的全局搜索能力；其次通过引入自适应权重因子来提高蝴蝶的局部搜索能力；最后采用动态切换概率 p 平衡算法局部搜索和全局搜索的比重，提升了算法的寻优性能。因此本文提出一种混合策略改进的蝴蝶优化算法(CWBOA)。

3.1 柯西变异

针对蝴蝶优化算法易陷入局部最优的特点，利用柯西变异来增加种群的多样性，提高算法的全局搜索能力，增加搜索空间。柯西分布函数在原点处的峰值较小但在两端的分布比较长，利用柯西变异

能够在当前变异的蝴蝶个体附近生成更大的扰动从而使得柯西分布函数的范围比较广^[11]，采用柯西变异两端分布更容易跳出局部最优值。本文融入柯西算子，充分利用柯西分布函数两端变异的效果来优化算全局最优个体，使得算法能够更好的达到全局最优。标准柯西分布函数公式如下。

$$f(x) = \frac{1}{\pi} \left(\frac{1}{x^2 + 1} \right) \quad (4)$$

柯西分布函数从峰值向两侧下降相对平缓，蝴蝶个体受局部的极值点约束力在进行柯西变异后下降，并且柯西分布函数的峰值相对较小，蝴蝶个体在变异后会使用相对较少的时间来搜索相邻区间，把更多的时间放在搜寻全局最优值上，使得改进的蝴蝶优化算法在寻找全局的最优值方面具备很好的调节能力。用柯西变异进行随机扰动有利于增加种群的多样性从而避免算法陷入局部最优，提高全局搜索最优值的能力。柯西分布的特征使其能够产生与原点相距较远的随机数，这意味着经过柯西变异后的蝴蝶个体具备了能够迅速逃离局部极值的能力。另外，柯西分布函数的峰值较低，该特点能够缩短变异后的蝴蝶个体在邻域周围搜索的时间。因此在求得当前最优解后，本文使用公式(5)所示的更新公式对当前全局最优解进行变异处理。

$$x_{newbest} = x_{best} + x_{best} * \text{Cauchy}(0,1) \quad (5)$$

3.2 自适应权重

惯性权重因子是很重要的一个参数，当惯性权重比较大时，算法全局搜索能力较强，能够增加种群多样性，可以搜索较大的区域；当惯性权重比较小时，算法局部搜索能力较强，可以在最优解周围精细搜索，加快收敛速度。蝴蝶进行局部寻优时，是以公式(3)进行局部搜索的，其中当蝴蝶以公式(3)向局部最优解靠近时，这时只能靠近局部最优解，而不能进行更好的局部寻优^[12]，针对这个问题受文献[13]，[14]的启发，提出了新的自适应权重方法，蝴蝶接近食物的时候，采用较小的自适应权重改变此时最优的蝴蝶的位置，使得蝴蝶局部寻优能力得

到提高。自适应权重公式如(6)所示，改进式(3)的公式如(7)所示。

$$w = \sin\left(\frac{\pi \cdot t}{(2 \cdot it \max)} + \pi\right) + 1 \quad (6)$$

式(6)中， t 为当前迭代次数， $it \max$ 是最大迭代次数。

$$x_i^{t+1} = w \cdot x_i^t + (r^2 \times x_j' - x_k') \times f_i \quad (7)$$

通过融合自适应权重因子 w ，使蝴蝶个体具有更好的局部寻优能力，改进后的蝴蝶算法流程图如图1所示。

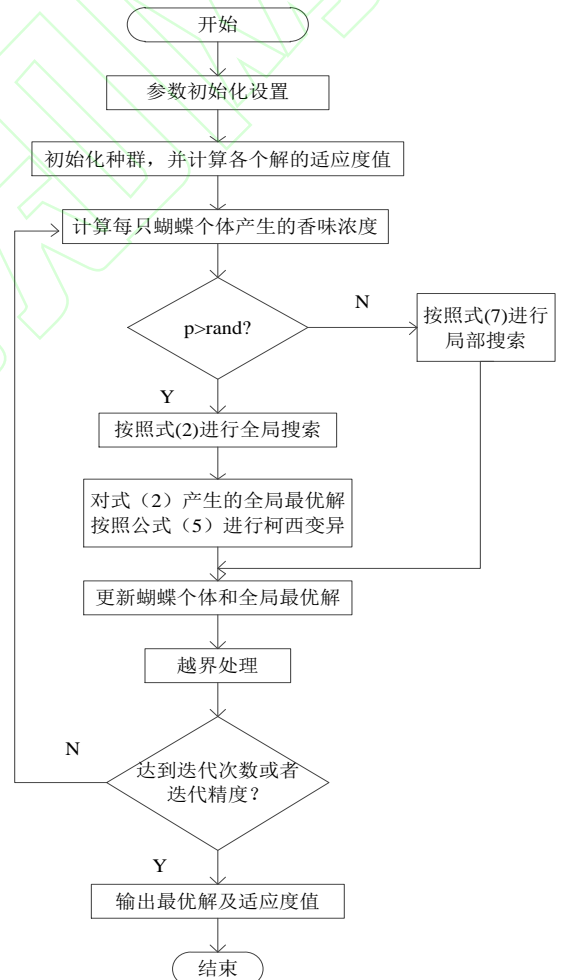


图1 改进算法的流程图

3.3 动态切换概率策略

在基本蝴蝶优化算法中，局部搜索和全局寻优过程用常量切换概率 $p \in [0,1]$ 来控制，一个合理的搜索过程在算法的前期应该进行较强烈全局搜

索，迅速定位搜索空间中全局最优解所在范围，在探索后期局部开采能力应增强，以提升算法的寻优精度。本文引入动态切换概率来平衡局部开采和全局开采的比重，来实现更好的寻优策略。动态切换概率 p 的公式如下。

$$p = 0.6 - 0.1 \times (\text{MaxIter} - t) / \text{MaxIter} \quad (8)$$

3.4 算法描述

CWBOA 的具体执行步骤如下：

步骤一：初始化种群及各个参数设置；

步骤二：计算每只蝴蝶的香味浓度，得到每个个体的适应度值，并且求出当前最优解；

步骤三：若动态转换概率 $p > \text{rand}$ ，则按照式(2)，(5)对全局位置进行更新；

步骤四：若动态转换概率 $p < \text{rand}$ ，则按照公式(7)进行局部位置更新；

步骤五：越界处理；

步骤六：计算步骤四和五的函数适应度值，若得到新的函数适应度值，则更新全局最优适应度值和全局最优解；

步骤七：判断结束条件，如果满足，则退出程序，输出最优值及最优解，否则转步骤三。

3.5 改进算法(CWBOA)的时间复杂度分析

设蝴蝶群体的规模为 N ，迭代次数为 MaxIter ，维度为 D ，则根据 CWBOA 算法的描述和时间复杂度符号 O 的运算规则，随机初始化种群的时间复杂度为 $O(N \cdot D)$ ，以及找到当前最香

的蝴蝶的时间的复杂度 $O(N \cdot D)$ ，利用柯西分布函数对于全局位置更新进行变异的时间复杂度为 $O(D)$ ，引入惯性权重因子对局部位置更新的时间复杂度为 $O(\text{MaxIter} \cdot D \cdot N)$ ，CWBOA 算法的总的时间复杂度为 $O(D \cdot N \cdot \text{MaxIter})$ ，所以 CWBOA 的时间复杂度与 BOA 相同，并没有增加计算负担。

4 函数测试与结果分析

4.1 仿真实验环境

本仿真测试环境为：操作系统 Windows7，CPU 为 Intel(R) Core(TM) i5-4210U，主频 1.7GHz，内存为 4GB，仿真软件为 Matlab2018b。

4.2 实验的初始参数设置

本文选取了基于柯西变异和动态自适应权重的蝴蝶优化算法(CWBOA)，基本蝴蝶算法(BOA)，鲸鱼算法(WOA)^[15]，以及花授粉算法(FPA)^[16]进行对比。为了实验的公平、客观性，本文将所有算法的初始种群规模统一设为 30，迭代次数设置为 500，四个算法的共有参数保持一致。CWBOA 和 BOA 中的 c 感官形态设置为 0.01， a 幂指数在迭代过程从 0.1 迭代到 0.3；基本的 BOA 和 FPA 中的切换概率均为 $p = 0.8$ 。

4.3 测试函数

为了验证改进后的 BOA 在收敛性和鲁棒性两方面的性能上更优，本文基于 14 个测试函数进行对比实验，标准测试函数的信息见表 2。

表 2 测试函数的基本信息

函数	名称	维数	范围	最优值
$f_1 = \sum_{i=1}^n x_i^2$	Sphere	30	[-100,100]	0
$f_2 = \max(x_i , 1 \leq i \leq n)$	Schwefel2.21	30	[-10,10]	0
$f_3 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i$	Schwefel2.22	30	[-10,10]	0
$f_4 = \sum_{i=1}^n ix_i^4 + rand(0,1)$	Quartic	30	[-1.28,1.28]	0
$f_5 = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	Matyas	30	[-10,10]	0
$f_6 = \sum_{i=1}^n ix_i^2$	Sum squares	30	[-5.12,5.12]	0
$f_7 = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	Bohachevsky	2	[-100,100]	0
$f_8 = \sum_{i=1}^d x_i^2 + (\sum_{i=1}^d 0.5ix_i)^2 + (\sum_{i=1}^d 0.5ix_i)^4$	Zakjarov	30	[-5,10]	0
$f_9 = \sum_{i=1}^{\frac{n}{4}} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	Powell	30	[-4,5]	0
$f_{10} = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	Apline	30	[-10,10]	0
$f_{11} = -20\exp(-0.2 \times \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	Ackley	30	[-32,32]	0
$f_{12} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	Griewank	30	[-600,600]	0
$f_{13} = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	Rastrain	30	[-5.12,5.12]	0
$f_{14} = 0.5 + \frac{\sin^2(\sqrt{x_i^2 + x_2^2} - 0.5)}{(1.0 + 0.001(x_i^2 + x_2^2))^2}$	Schaffter	2	[-100,100]	0

4.4 实验结果与分析

为了验证本文中改进 CWBOA 算法具有更好的收敛性和稳定性, 本文用 14 个基准测试函数进行了验证, 为了避免由于偶然的因素带来的结果的

偏差, 每个算法在每个测试函数上面独立运行 30 次。表 3 将给 CWBOA, BOA, BA, FPA 四个算法在多个测试函数上通过独立运行 30 次后所得到的实验结果。

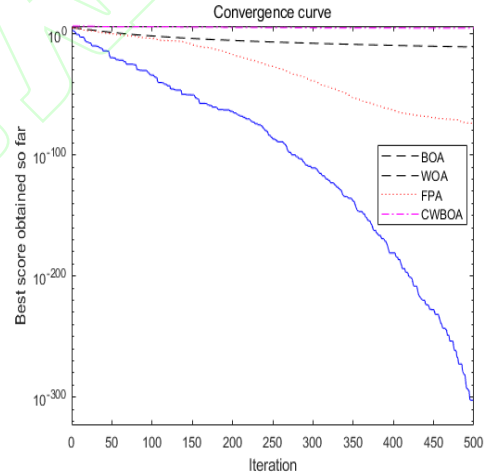
表 3 测试函数实验结果

函数	算法	最优值	最差值	平均值	标准差
f_1	CWBOA	0	0	0	0
	BOA	1.075E-11	1.4507E-11	1.2951E-11	1.0337E-12
	WOA	8.09E-87	2.70E-72	1.15E-73	4.8547E-73
	FPA	2.05E+04	6.50E+04	4.38E+04	1.07E+04
f_2	CWBOA	1.73E-149	9.87E-133	3.29E-134	1.80E-133
	BOA	5.38E-09	6.69E-09	6.69E-09	3.48E-10
	WOA	2.44E-01	8.48E+01	4.60E+01	2.49E+01
	FPA	7.39E-11	1.11E-06	1.53E-07	2.23E-07
f_3	CWBOA	3.18E-159	7.41E-133	3.86E-134	1.52E-133
	BOA	1.29E-09	5.69E-09	4.38E-09	1.38E-09
	WOA	3.13E-57	1.09E-48	3.85E-50	1.95E-49
	FPA	1.06E+03	2.22E+03	1.84E+03	2.44E+02
f_4	CWBOA	7.42E-06	5.05E-04	1.57E-04	1.35E-04
	BOA	6.53E-04	3.00E-03	1.41E-03	5.66E-04
	WOA	5.70E-05	1.93E-02	3.19E-03	3.94E-03
	FPA	7.53E-06	1.90E-03	7.00E-04	4.94E-04
f_5	CWBOA	0	0	0	0
	BOA	4.76E-13	1.39E-12	8.89E-13	2.27E-13
	WOA	2.55E-230	9.23E-185	3.08E-186	2.26E-185
	FPA	8.63E-10	9.39E-08	2.12E-08	2.53E-08
f_6	CWBOA	0	0	0	0
	BOA	3.31E-12	1.35E-11	1.20E-11	1.77E-12
	WOA	9.95E-89	6.35E-72	2.23E-73	1.14E-72
	FPA	2.87E+05	8.82E+05	5.53E+05	1.49E+05
f_7	CWBOA	0	0	0	0
	BOA	1.81E-13	4.14E-01	2.76E-02	1.05E-01
	WOA	0	0	0	0
	FPA	2.12E-06	2.63E-03	3.00E-04	5.03E-04
f_8	CWBOA	0	0	0	0
	BOA	9.15E-12	1.28E-11	1.10E-11	8.69E-13
	WOA	2.88E+02	6.59E+02	4.83E+02	9.41E+01
	FPA	1.94E+11	1.87E+12	6.68E+11	3.63E+11
f_9	CWBOA	0	0	0	0
	BOA	7.31E-12	1.23E-11	1.03E-11	1.16E-12
	WOA	1.71E-05	7.57E-05	4.27E-06	1.40E-05
	FPA	2.02E+05	4.70E+06	1.33E+06	9.56E+05
f_{10}	CWBOA	5.02E-156	4.28E-136	2.13E-137	8.15E-137
	BOA	1.30E-10	1.70E-09	6.10E-10	3.62E-10
	WOA	1.05E-58	3.16E-39	1.05E-40	5.67E-40
	FPA	5.80E+02	1.58E+03	1.13E+03	3.38E+02
f_{11}	CWBOA	8.88E-16	8.88E-16	8.88E-16	0

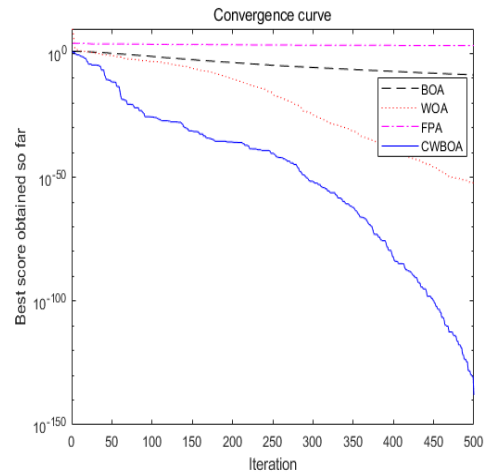
	BOA	4.45E-09	6.91E-09	6.05E-09	5.78E-10
	WOA	8.88E-16	7.99E-15	4.91E-15	2.55E-15
	FPA	4.09E+02	4.67E+02	4.43E+02	1.39E+01
	CWBOA	0	0	0	0
f_{12}	BOA	9.22E-13	9.84E-12	4.84E-12	2.26E-12
	WOA	4.51E+02	6.98E+02	6.01E+02	6.26E+01
	FPA	3.51E+01	5.57E+01	4.37E+01	5.66E+00
	CWBOA	0	0	0	0
f_{13}	BOA	0	2.18E+02	2.80E+01	6.78E+01
	WOA	0	0	0	0
	FPA	2.63E+04	7.42E+04	4.50E+04	1.10E+04
	CWBOA	0	0	0	0
f_{14}	BOA	5.32E-12	3.73E-02	1.15E-02	7.97E-03
	WOA	0	9.72E-03	6.48E-04	2.42E-03
	FPA	7.20E-03	9.80E-03	9.57E-03	5.25E-04

表 3, 分别列出了 CWBOA, BOA, WOA, FPA 算法独立运行 30 次所得到的最优值, 最差值, 平均值和标准差。我们很容易看出对于所选测试函数, CWBOA 的寻优性能最强, 明显优于 BOA, WOA, FPA。函数 $f_1, f_5, f_6, f_7, f_8, f_9, f_{12}, f_{13}, f_{14}$ 可以直接搜索到最优值 0, 寻优效果可以达到 100%; 对于函数 f_4 , CWBOA 的寻优平均值和基本花授粉算的寻优性能差不多, 略优于基本 BOA; 对于函数 f_{11} , CWBOA 的寻优精度平均值达到了 8.8818E-16 的水平, 比基本的蝴蝶优化算法提高了 7 个精度, 算法整体的稳定性比较好; 对于函数 f_{13} , CWBOA, BOA, WOA 均最优值均为 0, 但是在稳定性上面, CWBOA 的寻优性能要远远优于 BOA 和 WOA, 而 FPA 的寻优效果都是比较差的, 可见改进后的算法的性能具有明显的竞争优势。

为了直观展示 CWBOA 的寻优性能, 本文选取了 7 个基准测试函数的收敛迭代曲线图。这里仅给出 $f_1, f_3, f_7, f_9, f_{11}, f_{12}, f_{14}$ 的函数图像。如图 2 所示。



f_1 收敛曲线



f_3 收敛曲线

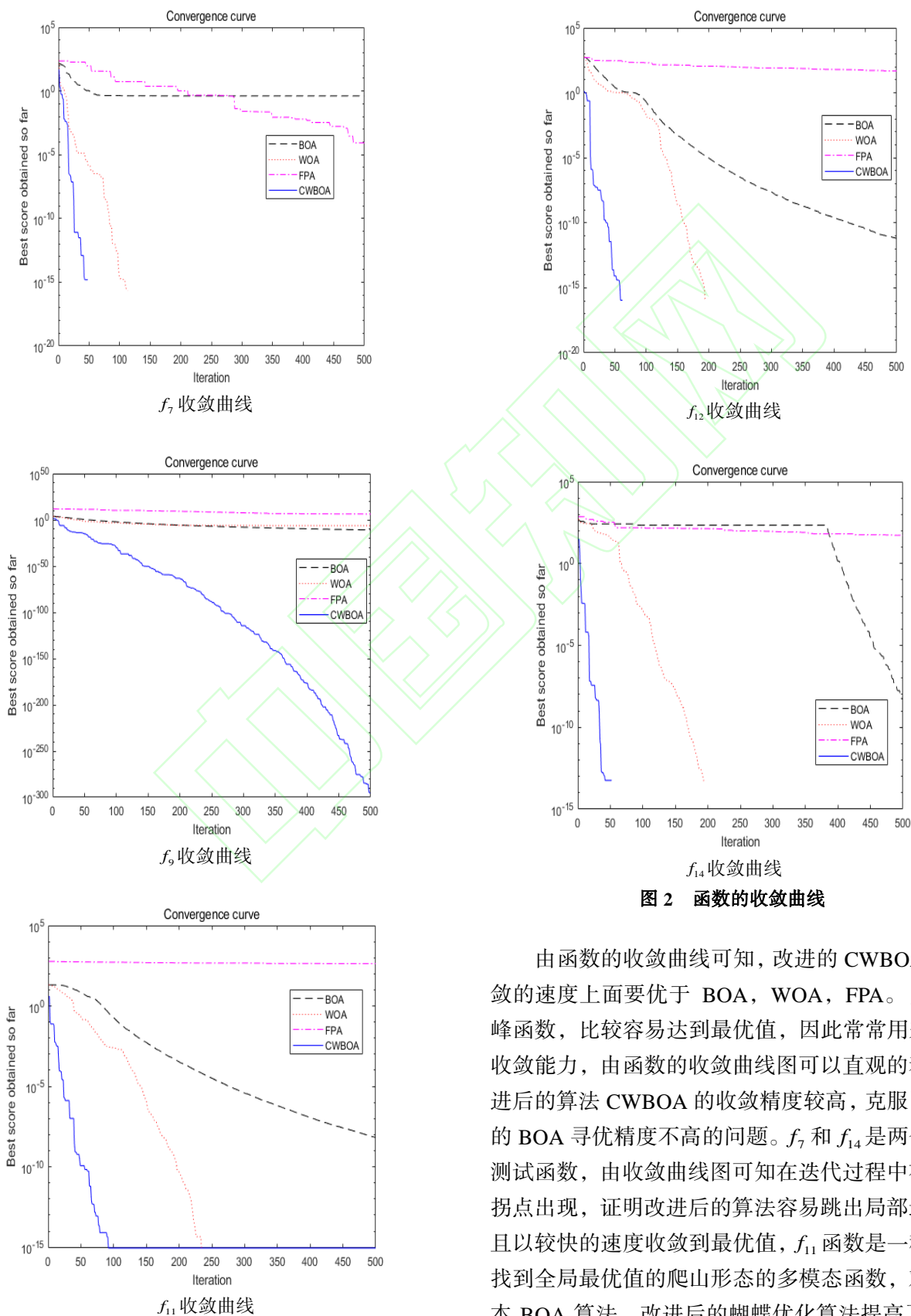


图 2 函数的收敛曲线

由函数的收敛曲线可知,改进的 CWBOA 在收敛的速度上面要优于 BOA, WOA, FPA。 f_1 是单峰函数,比较容易达到最优值,因此常常用来测试收敛能力,由函数的收敛曲线图可以直观的看出改进后的算法 CWBOA 的收敛精度较高,克服了基本的 BOA 寻优精度不高的问题。 f_7 和 f_{14} 是两个低维测试函数,由收敛曲线图可知在迭代过程中有多个拐点出现,证明改进后的算法容易跳出局部最优并且以较快的速度收敛到最优值, f_{11} 函数是一种很难找到全局最优值的爬山形态的多模态函数,对比基本 BOA 算法,改进后的蝴蝶优化算法提高了 7 个

数量级的寻优精度。 f_{12} 是一个复杂的多峰函数有许多广泛的局部最小值,并且它们是有规律地分布的,在(0, 0)处取得最优值 0, CWBOA 在不到 50 代已经达到最优值而且寻优速度也远远快于 BOA, WOA 和 FPA, 由收敛曲线图可以知, CWBOA 的收敛性要优于基本的 BOA, WOA, FPA。

另外, 本文的改进算法与文献[7]中的 BA, 文献[8]中 LBOA, 文献[9]中 IBOA 中的分别进行比较,

每个算法的初始化种群数均为 30, 其中 BA 和 LABOA 的最大迭代次数为 1000, CWBOA 和 IBOA 的最大迭代次数为 500, 均参考原文献实验参数设置, 本文设置最大迭代次数为 500 次的实验结果已经明显优于文献[7][8][9]中的对比算法的实验结果了。根据实验结果可知本文改进算法的寻优效果更佳。对比结果如表 4 所示。“/”表示原文献的实验中不涉及的测试函数。

表 4 改进算法的实验结果对比

函数名	CWBOA		BA		LABOA		IBOA	
f_1	Mean	0	Mean	0	Mean	1.03E-11	Mean	1.90E-104
	Std	0	Std	0	Std	5.38E-13	Std	6.01E-104
f_2	Mean	3.29E-134	Mean	3.71E-177	Mean	4.62E-09	Mean	4.51E-10
	Std	1.80E-133	Std	0	Std	2.40E-10	Std	8.19E-11
f_3	Mean	3.86E-134	Mean	0	Mean	4.14E-09	Mean	/
	Std	1.52E-133	Std	0	Std	2.17E-10	Std	/
f_4	Mean	1.57E-04	Mean	4.90E-06	Mean	5.48E-04	Mean	1.59E-03
	Std	1.35E-04	Std	5.17E-04	Std	4.20E-04	Std	4.36E-04
f_5	Mean	0	Mean	/	Mean	/	Mean	1.11E-15
	Std	0	Std	/	Std	/	Std	2.38E-15
f_6	Mean	0	Mean	/	Mean	9.57E-12	Mean	/
	Std	0	Std	/	Std	6.09E-13	Std	/
f_7	Mean	0	Mean	/	Mean	/	Mean	1.71E+00
	Std	0	Std	/	Std	/	Std	0
f_{10}	Mean	2.13E-137	Mean	0	Mean	3.37E-09	Mean	/
	Std	8.15E-137	Std	0	Std	1.98E-10	Std	/
f_{11}	Mean	8.88E-16	Mean	/	Mean	4.29E-09	Mean	/
	Std	0	Std	/	Std	1.82E-10	Std	/
f_{12}	Mean	0	Mean	/	Mean	1.24E-11	Mean	1.44E-12
	Std	0	Std	/	Std	9.58E-13	Std	2.12E-13
f_{13}	Mean	0	Mean	0	Mean	1.15E-11	Mean	/
	Std	0	Std	0	Std	2.22E-12	Std	/
f_{14}	Mean	0	Mean	0	Mean	/	Mean	/
	Std	0	Std	0	Std	/	Std	/

4.5 求解高维函数的实验分析

根据上述低维实验结果对比分析可知, 本文的改进算法在低维测试函数上无论是最优值还是标准差上面均得到了较好的收敛效果。但是一般的改进策略在较为复杂的工程问题上面, 特别是高维测试

函数上极易失效, 为了测试本文改进算法的有效性, 进行了 CWBOA 的高维函数优化求解实验, 因为在实际的问题中普遍存在的是大规模的, 复杂的优化问题。为了更好的证明本文改进算法能够应用于求解大规模复杂问题, 高维参数设置与 4.2 节相同。

表 5 CWBOA 的高维函数求解结果

函数名	CWBOA						BOA
	D=30		D=100		D=500		D=500
	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度
1	0	0	0	0	0	0	1.16E-11
2	1.73E-149	1.80E-133	3.02E-149	6.03E-130	1.02E-147	5.03E-130	1.16E-09
3	3.18E-159	1.52E-133	-	-	-	-	-
4	7.42E-06	1.35E-04	1.40E-04	6.60E-04	2.50E-03	1.60E-03	4.23E-03
5	0	0	0	0	0	0	2.96E-12
6	0	0	0	0	0	0	3.21E-11
8	0	0	0	0	0	0	6.05E-10
9	0	0	0	0	0	0	8.31E-11
10	5.02E-156	8.15E-137	1.89E-136	4.24E-136	1.09E-133	5.24E-130	5.30E-11
11	8.88E-16	0	8.88E-16	0	8.88E-16	0	4.82E-12
12	0	0	0	0	0	0	1.52E-11
13	0	0	0	0	0	0	0

根据表 5 的实验结果,我们发现 CWBOA 求解高维函数上面的效果比较好,但是在解决 100 维, 500 维的 f_3 函数的时候失效。 f_3 是连续的,平滑的多峰函数,当自变量趋近于无穷大时,函数会形成大量局部极值区域,易陷入局部最优且不易跳出,在高维测试函数下求解难度较大,在测试函数维数大于 100 维时,就属于大规模函数优化问题。随着搜索空间维数的增加,问题的复杂度以及指数级数的增长,从而出现“维数灾难”问题^[17],根据无万能算法理论,没有任何一个算法适合解决所有问题,所以这个结果是可以接受的。函数 f_1 , $f_5 \sim f_9$, f_{12} , f_{13} 均能达到最优值 0,所以在求解高维测试函数时,本文的改进策略依然具有较强的稳定性,能够有效的处理复杂的高维问题。选取本文的测试函数 f_1 , f_4 , f_6 , f_{10} , f_{11} 测试结果与文献^[18]中的对立学习策略的鲸鱼优化算法(IWOA)在 500 维的情况下的结果进行对比(实验结果如表 6 所示)。函数 f_1 , f_6 这两个测试函数的结果始终能精确达到最优值 0,寻优效果能达到 100%,说明 CWBOA 在处理复杂的大规模问题上鲁棒性较强;对于函数 f_4 和 f_{11} ,CWBOA 的寻优精度略优于 IWOA,但是

标准差始终稳定在 0,表明其较强的寻优稳定性,对于函数 f_{10} ,本文的改进算法要比 IWOA 的寻优精度提高近 70 个单位。综上,能够证明本文改进算法的可行性和有效性。

表 6 CWBOA 与 IWOA 对比结果

初始化参数设置相同 D=500, N=30, $MaxIter=500$				
函数名	CWBOA		IWOA	
	平均精度	标准差	平均精度	标准差
f_1	0	0	4.85E-113	3.28E-113
f_4	1.40E-04	6.60E-04	2.04E-03	2.47E-03
f_6	0	0	7.88E-113	4.90E-113
f_{10}	1.89E-136	4.24E-136	1.25E-67	1.69E-67
f_{11}	8.88E-16	0	4.44E-15	0

5 结束语

本文提出了一种混合策略的蝴蝶优化算法(CWBOA),在全局位置更新后的值引入柯西变异算子,增加了种群多样性,降低算法陷入局部最优的可能性;在 BOA 局部位置更新中引入自适应权重因子,提高了算法的局部搜索能力,并且采用动态切换概率 p 来权衡局部开采和全局搜索的比重,基于三种改进策略提升了算法的寻优能力和鲁棒性。接下来的研究方向主要是拓展改进算法的实际应用领域。

参考文献:

- [1] Deb S, Deb S, Gao X Z, et al. A new metaheuristic optimisation algorithm motivated by elephant herding behaviour[J]. *International Journal of Bio-Inspired Computation*, 2017, 8(6):394-409.
- [2] Gandomi A H, Alavi A H. Krill herd: A new bio-inspired optimization algorithm[J]. *Communications in Nonlinear Science&Numerical Simulation*, 2012, 17(12):4831-4845.
- [3] Mirjalili S. The Ant Lion Optimizer[J]. *Advances in Engineering Software*, 2015, 83(C):80-98.
- [4] Askarzadeh, Alireza. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm[J]. *Computers & Structures*, 2016, 169:1-12.
- [5] Meng X, Liu Y, Gao X, et al. A New Bio-inspired Algorithm: Chicken Swarm Optimization[C]//*International Conference in Swarm Intelligence*. Springer International Publishing, 2014,8794:86-94.
- [6] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. *Soft Computing*, 2019, 23(3): 715-734.
- [7] Arora S, Singh S. Butterfly algorithm with Lèvy Flights for global optimization[C]//*2015 International Conference on Signal Processing, Computing and Control (ISPPC)*. IEEE, 2015: 220-224.
- [8] Sankalap A, Priyanka A. Learning automata based butterfly optimization algorithm for engineering design problems[J]. *International Journal of Computational Materials Science and Engineering*, 2018,7(4): 1850021.
- [9] Arora S, Singh S. An Improved Butterfly Optimization Algorithm for Global Optimization[J]. *Advanced Science*, 2017, 8(9):711-717.
- [10] Arora, S. and Singh, S. An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization[J], *International Journal of Interactive Multimedia and Artificial Intelligence*, 2017, 4(4): 14-21.
- [11] 郭振洲, 王平, 马云峰, 等. 基于自适应权重和柯西变异的鲸鱼优化算法[J]. *微电子学与计算机*, 2017, 34(9): 20-25.
- [12] 贺智明, 李文静. 基于动态全局搜索和柯西变异的花授粉算法[J/OL]. *计算机工程与应用*:1-13. [2019-08-11]. <http://kns.cnki.net/kcms/detail/11.2127.tp.20181121.1207.008.html>.
- [13] 赵志刚, 黄树运, 王伟倩. 基于随机惯性权重的简化粒子群优化算法[J]. *计算机应用研究*, 2014, 30(2): 361-363.
- [14] 党婷婷, 林丹. 含有动态自适应惯性权重的蜘蛛猴优化算法[J]. *计算机工程与应用*, 2019,55(14):40-47.
- [15] Mirjalili S, Lewis A. The Whale Optimization Algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [16] Yang X S. Flower Pollination Algorithm for Global Optimization[C]//*International Conference on Unconventional Computation and Natural Computation*. Berlin: Springer Verlag, 2013: 240-249.
- [17] Yong W, Cai Z, Zhou Y, et al. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique[J]. *Structural & Multidisciplinary Optimization*, 2009, 37(4):395-413.
- [18] 龙文, 蔡绍洪, 焦建军, 等. 求解大规模优化问题的改进鲸鱼优化算法[J]. *系统工程理论与实践*, 2017, 37(11): 2983-2994.