计算机应用研究
Application Research of Computers
ISSN 1001-3695,CN 51-1196/TP

《计算机应用研究》网络首发论文

题目：全局优化的蝴蝶优化算法
作者：高文欣，刘升，肖子雅，于建芳
DOI：10.19734/j.issn.1001-3695.2019.07.0274
收稿日期：2019-07-31
网络首发日期：2019-12-26
引用格式：高文欣，刘升，肖子雅，于建芳．全局优化的蝴蝶优化算法[J/OL]．计算机应用研究. https://doi.org/10.19734/j.issn.1001-3695.2019.07.0274

网络首发时间：2019-12-26 10:57:50
网络首发地址：http://kns.cnki.net/kcms/detail/51.1196.TP.20191225.1401.021.html

全局优化的蝴蝶优化算法 *

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中存在的易陷入局部最优值，收敛速度慢等问题，提出一种全局优化的蝴蝶算法，引入 limit 阈值来限定蝴蝶优化算法陷入局部最优解的次数，从而改变算法易陷入早熟的问题，结合单纯形策略优化迭代后期位置较差的蝴蝶使种群能够较快的找到全局最优解；将正弦余弦算法作为局部算子融入 BOA 中，改善迭代后期种群多样性下降的缺陷，加快算法跳出局部最优。在仿真模拟实验中与多个算法进行对比，结果表明改进算法的寻优性能更好。

关键词：蝴蝶优化算法；limit 阈值；单纯形法；正弦余弦算法

中图分类号：TP18　　doi: 10.19734/j.issn.1001-3695.2019.07.0274

Butterfly optimization algorithm for global optimization

Gao Wenxin, Liu Sheng†, Xiao Ziya, Yu Jianfang

(*College of Management*, *Shanghai University of Engineering Sciences*, *Shanghai* 201620, *China*)

Abstract: Aiming at the problems of easy falling into local optimum and slow convergence speed in basic butterfly optimization algorithm, proposing a global optimization butterfly algorithm. Limit threshold is introduced to limit the number of times butterfly optimization algorithm falls into local optimum solution, so as to change the problem that the algorithm is easy to fall into premature. Simple strategy is combined to optimize iteration. For the butterfly individuals with poor position in the later stage, the population can converge to the global optimum faster. The sine-cosine algorithm is used as a local operator to improve the shortcomings of population diversity decline in the later stage of iteration and accelerate the algorithm to jump out of the local optimum. Compared with several algorithms in the simulation experiment, the results show that the improved algorithm has better optimization performance.

Key words: butterfly optimization algorithm; limit threshold; simplex method; sine-cosine algorithm

0　引言

蝴蝶优化算法(butterfly optimization algorithm)[1]是 Arora 和 Singh 两位学者提出的一种模拟蝴蝶觅食和求偶行为的新型元启发式优化算法。基本 BOA 中参数少、原理简单、易于实现。在研究中，发现 BOA 的收敛速度和寻优精度要明显 DE(差分进化算法)[2]，CS(布谷鸟搜索算法)[3]，PSO(粒子群优化算法)[4]，FA(萤火虫算法)[5]，GA(遗传算法)[6]，ABC(人工蜂群算法)[7]。但是基本的蝴蝶优化算法中也存在收敛精度低和易陷入局部最优的问题。

国外的部分学者进行了一些有效的改进 BOA 算法。2015 年，Arora 和 Singh 两位学者年提出了一种基于莱维飞行策略的蝴蝶优化算法[8]，在全局和局部的两个位置更新处引入莱维函数，增加了种群的多样性，提高了算法的全局开采能力；2018 年，Arora 和 Singh 两位学者提出了一种基于学习自动机机制的蝴蝶优化算法[9]，引入学习自动机机制，在保持学习自动机的主要特征的同时，加速了全局收敛速度。文献[10]提出了一种基于感觉模态变化的蝴蝶优化算法，采用变量感知模态参数策略，提高了收敛速度；文献[11]提出了一种融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。虽然已有学者的研究对基本 BOA 算法的寻优能力有所提高，但对于权衡算法全局开采和局部搜索能力、解决算法收敛速度慢、易陷入局部最优等问题仍需更加深入的研究。

本文针对基本蝴蝶优化算法中存在的寻优精度不高，收敛速度慢的缺陷，提出了一种全局优化的蝴蝶优化算法,在局部探索过程中将正弦余弦算法作为一种局部优化算子融入 BOA 中，使其能在算法迭代后期较快的跳出局部最优；在全局搜索阶段，本文参照人工蜂群算法具有较强的全局搜索能力，将人工蜂群算法中蜜蜂抛弃蜜源的行为引入 BOA 算法中，结合 limit 阈值的策略，如果在连续 limit 代寻优过程中某个解没有被改善，那么将抛弃这个解而重新随机生成一个解进入下一代的进化[12]，另外结合单纯形法来优化在迭代过程中位置较差的蝴蝶，使种群能够较快的收敛到全局最优解，使得算法能够有效克服易陷入局部最优解的缺陷、增强算法的全局搜索能力、避免早熟，提升了算法的寻优精度和收敛速度。

1　蝴蝶优化算法

蝴蝶优化算法是模拟自然界中蝴蝶觅食(花蜜)和求偶行为而衍生出的一种新型群智能优化算法。从生物学上讲，蝴蝶有感官感受器用来闻/感觉食物/花朵的香味。这些感受器被称为化学感受器，分散在蝴蝶身体的各个部位，如腿、触须等。在 BOA 中，假设每只蝴蝶都能产生不同程度的香味。这种香味与蝴蝶的适应性有着进一步的关系。蝴蝶在搜索空间中从一个位置移动到另一个特定位置，它的适应度就会相

收稿日期：2019-07-31；**修回日期：**2019-09-03　**基金项目：**国家自然科学基金资助项目(61075115，61673258)；上海市自然科学基金资助项目(19ZR1421600)

作者简介：高文欣(1995-)，女，河北承德人，硕士研究生，主要研究方向为群智能计算、智能计算、大数据处理与分析；刘升(1966-)，男(通信作者)，湖北黄石人，教授，博士，主要研究方向为人工智能，智能计算(ls6601@163.com)；肖子雅(1994-)，女，江苏徐州人，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，河南周口人，硕士研究生，主要研究方向为群智能算法、智能计算，工程优化，项目调度.

应变化。蝴蝶所产生的香味被远远地传播到该地区的所有其他蝴蝶身上，其他蝴蝶感受到了传播的香味就会形成了一个群体的社会知识网络。每当蝴蝶在搜索空间中感受到来自最优蝴蝶的香味时，它就会朝着最优蝴蝶移动，这个阶段被称为 BOA 的全局探索阶段，当蝴蝶无法在搜索空间中感受到任何其他蝴蝶的香味时，它将随机游走，这个阶段称为 BOA 中的局部开采阶段。针对上述行为，给出如下模拟步骤。

a)所有的蝴蝶会散发香味，使蝴蝶个体之间相互吸引。

b)每只蝴蝶都会随机移动或向发出最多香味的蝴蝶移动。

c)蝴蝶的刺激强度受目标函数的影响或决定。

d)全局搜索和局部搜索使用切换概率 P 来控制，由于物理近似度以及风雨，雷电等各种其他自然等因素，局部搜索和全局开采的切换概率 P 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，同时这也是区别于其他的群智能算法的一个重要特征。蝴蝶产生香味的数学公式如下。

$$f = cI^{a} \tag{1}$$

其中 f 是香味的感知强度，即香味能够被其他蝴蝶感知的强度，c 是蝴蝶的感觉模态，I 是刺激强度，a 是依赖于模态的幂指数，它解释了不同程度香味的吸收。对于现实中的大多数情况，可以在[0,1]范围内取 a。参数 a 是取决于感觉模态的功率指数(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式(1)计算在它们的位置处产生的香味。在全局搜索阶段，蝴蝶朝着最优的蝴蝶(解 g^{*})移动，全局位置更新用式(2)表示。

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

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 只蝴蝶发出的香味用 f_i 表示，r 是[0,1]之间的随机数。局部位置更新公式如下。

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

x_j^t 和 x_k^t 表示解空间中的第 j 只和第 k 只蝴蝶的解向量，若 x_j^t 和 x_k^t 属于同一个群体，并且 r 是[0,1]内的随机数，表示局部随机游走。基本蝴蝶的算法流程图如图 1 所示。

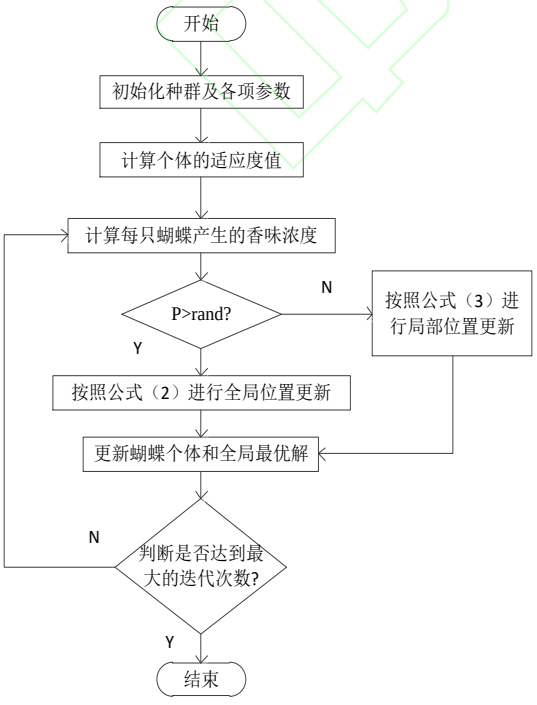


图 1　基本 BOA 算法流程图

Fig. 1　Flow chart of basic BOA algorithm

2　全局优化的蝴蝶优化算法

本文针对基本蝴蝶优化算法中存在的寻优精度不高，收敛速度慢的缺陷，提出了一种混合策略改进的蝴蝶优化算法 simplex method sine-cosine algorithm butterfly optimization algorithm(SMSCABOA)，在局部探索过程中将正弦余弦算法作为一种局部优化算子融入 BOA 中，使其能较快的跳出局部最优；在全局搜索阶段，本文参照人工蜂群算法其有较强的全局搜索能力，将人工蜂群算法中蜜蜂抛弃蜜源的行为引入 BOA 中，结合 limit 阈值的策略，如果在连续 limit 代寻优过程中某个解没有被改善，那么将抛弃这个解而重新随机生成一个解进入下一代的进化，本文使用单纯形法来产生新的解。改进策略使得算法能够有效跳出局部最优、增强算法的全局搜索能力、避免早熟，提升了算法的寻优精度和收敛速度。

2.1　limit 阈值

本文参考人工蜂群算法[13]中侦察蜂阶段引入了观察可行解的停滞次数 limit 阈值从而跳出局部最优，较快的寻得全局最优解的思想，引入 limit 阈值来克服算法易陷入局部最优问题。limit 阈值机制是在预先设定极限值(limit)的条件下，通过观察可行解的停滞次数是否达到 limit 值来进行循环迭代，若在 limit 代中某个个体的位置没有改变，则抛弃当前解，重新生成下一个解进行下一代的进化。

通过分析基本 BOA 算法可知，随着算法的迭代次数的增加，蝴蝶个体将逐渐向适应度较优的个体飞行从而使得种群分布缩小，种群的多样性呈现下降趋势，导致算法陷入局部最优。因此引入 limit 阈值来限定蝴蝶优化算法陷入局部最优解的次数，从而改变算法易陷入早熟的问题。阈值 limit 的取值会影响算法寻优能力，若取值过大则对算法改进甚微；反之，取值过小则会导致算法的局部开发能力被减弱，且算法更新频繁。通过多次测试，本文将阈值 limit 设定为 60 效果比较好。

2.2　单纯形法

Nelder-Mead 单纯形法[14]是由 Nelder 和 Mead 两位学者在 1965 年提出的一种用于优化多维无约束问题的搜索算法。此算法通过在 d 维空间中取个 $d+1$ 点构成一个单纯形，首先计算顶点的函数值；然后对单纯形的最差点进行内部压缩、外部压缩、反射和扩展等操作，来获得更好的点，由此替换最差点，重构单纯形，使其不断的逼近最优点[15]。单纯形算法是一种直接搜索算法，其寻优不受目标函数的连续性与可导性影响，具备精细的寻优能力。因此本文引入单纯形的搜索策略，在寻香结束后，选择位置较差的蝴蝶利用单纯形法来优化较差蝴蝶的位置，产生新的解，进而加快种群收敛到最优解的速度。单纯形法的步骤如图 2 所示。

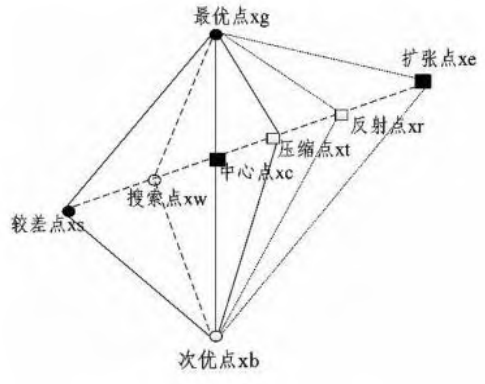


图 2　单纯形搜索的不同点

Fig. 2　The differences of simple search

具体过程如下。

a)评价蝴蝶种群中所有个体，从中选出最优的蝴蝶 x_g 和次优的蝴蝶 x_b，设 x_s 是要被抛弃的蝴蝶，x_g，x_b 和 x_s 的适应度值分别是 $f(x_g)$、$f(x_b)$ 和 $f(x_s)$。

b)计算出 x_g 和 x_b 的中点，记为 x_c。

$$x_c = \frac{x_g + x_b}{2}$$

c)执行反射操作。

$$x_r = x_c + a(x_c - x_s)$$

x_r 是根据 x_s 得到的反射点，a 是反射系数，通常取 1。

d)若 $f(x_g) > f(x_r)$，则进行扩张操作，得到扩张点 x_e

$$x_e = x_c + \gamma(x_r - x_c)$$

本文中扩张系数 γ =2。

若 $f(x_e) < f(x_g)$,就用 x_e 替换 x_s，否则用 x_r 替换 x_s。

e)若 $f(x_r) > f(x_s)$，进行压缩操作，得到压缩点 x_t。

$$x_t = x_c + \beta(x_s - x_c)$$

本文中 β =0.5。若 $f(x_t) < f(x_s)$,则用压缩点 x_t 替换 $\boldsymbol{x_s}$。

f)若 $f(x_s) > f(x_r) > f(x_g)$,进行收缩。

$$x_w = x_c + \beta(x_s - x_c)$$

x_w 是收缩点，收缩系数 β =0.5，$f(x_w) < f(x_s)$ 则用 x_w 替换 x_s，否则用 x_r 替换 x_s。

2.3　正弦余弦指引机制

本文引入正弦与余弦算法[16]，在算法的迭代后期，让所有的蝴蝶个体进行正弦余弦操作，来优化蝴蝶个体进行位置更新。具体的操作方式如下。

$$x_i^{t+1} = \begin{cases} x_i^t + R_1 \times \sin(R_2) \times |R_3 \times M_i^t - x_i^t| & R_4 \le 0.5 \\ x_i^t + R_1 \times \cos(R_2) \times |R_3 \times M_i^t - x_i^t| & R_4 > 0.5 \end{cases} \tag{4}$$

式(4)中含有四个参数，R_1，R_2，R_3，R_4，R_1 决定在下一次迭代第 i 个个体的位置更新方向，R_1 的范围影响着搜索空间的范围，$R_1 = a - T \times \frac{a}{T_{\max}}$，迭代次数的变化会缩小蝴蝶个体的搜索范围，有效的保证了算法的收敛性，使其最终收敛到一个最优值。a 是一个常数，本文中 $a=2$；R_2 是 $[0,2\pi]$ 之间的随机数，它决定下一次迭代中个体的移动距离；R_3 是随机权重的取值范围是[0,2]，$R_3>1$ 时，M_i^t 对下一代迭代中个体的位置更新具有明显的影响，否则没有影响效果；R_4 是 [0,1] 之间产生的一个随机数，R_4 决定蝴蝶个体的位置更新方式是正弦还是余弦操作。

在蝴蝶进行局部寻优时，引入正弦余弦算法进行个体位置更新操作，这种位置更新策略可以改善蝴蝶个体与最优个体之间的信息交流机制，避免寻优个体的盲目性，克服算法在迭代后期容易陷入局部最优的缺陷。在这个过程中，参数 R_1，R_2 可以有效的控制搜索的距离和方向。正弦操作能够有效的保证全局寻优得到潜在的最优解，来减少余弦搜索的盲目性，降低个体陷入局部最优的可能性，余弦操作可以促进迭代后期的局部搜索能力，进一步弥补迭代后期寻优速度下降的问题，有效的保证了寻优的效率。正余双弦相互作用，共同提高算法的寻优性能。

2.4　改进算法的时间复杂度分析

算法的种群规模设置为 N,最大迭代次数为 Maxiter,搜索空间的维度为 Dim，增加 limit 阈值策略，相当于增加一个内层循环(由改进算法的流程图可以直接看出)，则增加的运算量为 $O(\text{Maxiter} \times \text{Dim} \times (N-1))$，增加单纯形策略优化全局最优值其时间复杂度为 $O(\text{Maxiter} \times \text{Dim} \times N)$，增加正弦余弦指引机制其时间复杂度为 $O(\text{Maxiter} \times \text{Dim} \times N)$，因此，SMSCABOA 算法总的时间复杂度为 $O(\text{Maxiter} \times \text{Dim} \times (N-1))$ 相当于增加了一个内层循环，根据改进算法的寻优效率来说，增加的计算量是可以接受的。

2.5　改进算法的流程

本文改进算法的流程图如图 3 所示。

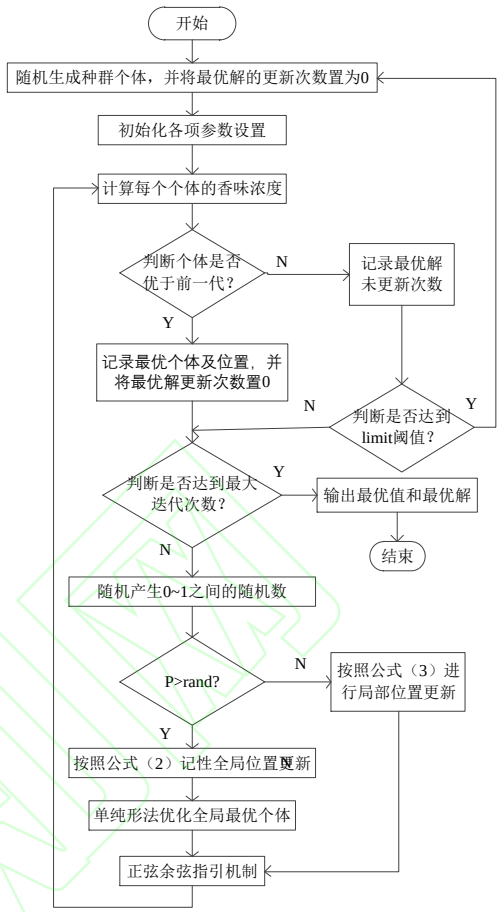


图 3　SMSCABOA 算法流程图

Fig. 3　Flow chart of SMSCABOA algorithm

3　仿真实验结果与分析

3.1　仿真实验环境

本仿真测试环境为：操作系统 Windows 7，CPU 为 Intel® Core™ i5-4210U，主频 1.7 GHz，内存为 4 GB，仿真软件为 MATLAB2018b。

3.2　实验的初始参数设置

本文选取樽海鞘群优化算法(SSA)[17]，鲸鱼优化算法(WOA)[18]，蚁狮优化算法(ALO)[19],基本蝴蝶蝴蝶优化算法(BOA),单一策略改进的正弦指引机制的蝴蝶优化算法(SCABOA),以及本文的全局优化的蝴蝶优化算法(SMSCABOA)进行对比实验，为了使实验公平、客观，本文将所有算法的种群规模统一设为 30，迭代次数设置为 500，四个算法的公有参数保持一致。BOA，SCABOA 和 SMSCABOA 的感官模态 c 设置为 0.01，a 幂指数在迭代过程从 0.1 迭代到 0.3，切换概率 p=0.8。SMSCABOA 中的 limit 设置为 60 效果较好。

3.3　测试函数

表 1 和 2 分别给出了本文选取的单模态和多模态测试函数。

表 1　单模态基准测试函数

Tab. 1　Single-mode benchmark function

函数	Dim	范围	fmin
$f_1 = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} x_i$	30	[-10,10]	0
$f_3 = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$	30	[-10,10]	0

表 2　多模态基准测试函数

Tab. 2　Multimodal benchmark function

函数	Dim	范围	fmin
$f_4=\frac{\pi}{d}\cdot\{10\sin^2(\pi y_1)\sum_{i=1}^{d-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{d}u(x_i,10,100,4)$	30	[-50,50]	0
$f_5=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	30	[-10,10]	0
$f_6=-20\exp(-0.2\times\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	[-32,32]	0
$f_7=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]	0
$f_8=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	30	[-5.12,5.12]	0
$f_9=0.5+\frac{\sin^2(\sqrt{x_1^2+x_2^2}-0.5)}{(1.0+0.001(x_1^2+x_2^2))^2}$	2	[-5.12,5.12]	0

3.4　实验结果及分析

为了验证本文中 SMSCABOA 算法具有更好的收敛性、稳定性，本文选用 9 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 SSA，WOA，ALO，BOA，SCABOA，SMSCABOA 六个算法在每个测试函数上，独立运行 30 次后所得到的实验结果。

在本文所选的测试函数中 f_1～f_3 是单峰函数，f_4 ～f_9 是复杂的非线性多峰函数。对于单模态测试函数，一般来说比较容易达到最优值，所以用来测试算法的收敛性能。本文的改进算法能够较快的收敛到最优值。而多模态测试函数一般会存在大量的局部极值，用来检验算法的全局开采能力、群体多样性以及跳出局部最优值和避免易陷入早熟的能力。从表 3 的实验结果可以看出，在固定迭代次数的实验下改进后的 SMSCABOA 的性能无论在单模态函数还是多模态函数上均优化明显，明显优于对比算法 SSA,WOA,ALO,BOA 的，除函数 f_4 ，f_6 这两个函数之外，其余测试函数都能收敛到最优值 0，改进算法的稳定性较好，寻优成功率达到了 100%。对于函数 f_4 虽然在迭代截止的时候没有收敛到最优值，终止精度达到了 10^{-9} 并且仍然有下降的趋势，但是其收敛精度相比于基本的 BOA 提高了 8 个精度单位，对比其他的四个算法，其寻优性能也是最好的。函数 f_6 是一种很难找到全局最优值的爬山形态的多峰函数，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，标准差为 0，证明改进算法具有较好的稳定性。

为了进一步验证本文使用混合策略改进算法的有效性，在表 3 中展示了使用单一策略正弦余弦指引机制优化的蝴蝶优化算法，从实验结果上可以看出，使用单策略改进的 SCABOA 算法在函数 f_1，f_6，f_7，f_8，f_9 上能够收敛到最优值，由此说明，正弦余弦指引机制能够有效的降低基本 BOA 算法在迭代过程中陷入局部最优值的可能性，避免蝴蝶个体盲目寻优。函数 f_2，f_3 不能收敛到最优值，其寻优精度达到了 E^{-220} 数量级，比基本蝴蝶算法提高 200 个精度，寻优成功率达到 67%,也进一步说明增加 limit 阈值和单纯形策略改进后的 SMSCABOA 算法具有更高的寻优精度，和鲁棒性，其结果是明显优于单一策略的。对于函数 f_2，SMSCABOA 算法比 SCABOA 算法在平均精度上面高出 4 个数量级，SMSCABOA 算法的标准差更小，稳定性更好。标准差一般是用来检验算法的稳定性，在实验结果中可以看出本文的 SMSCABOA 的标准差最小，可以证明混合改进后算法的稳定性更高。

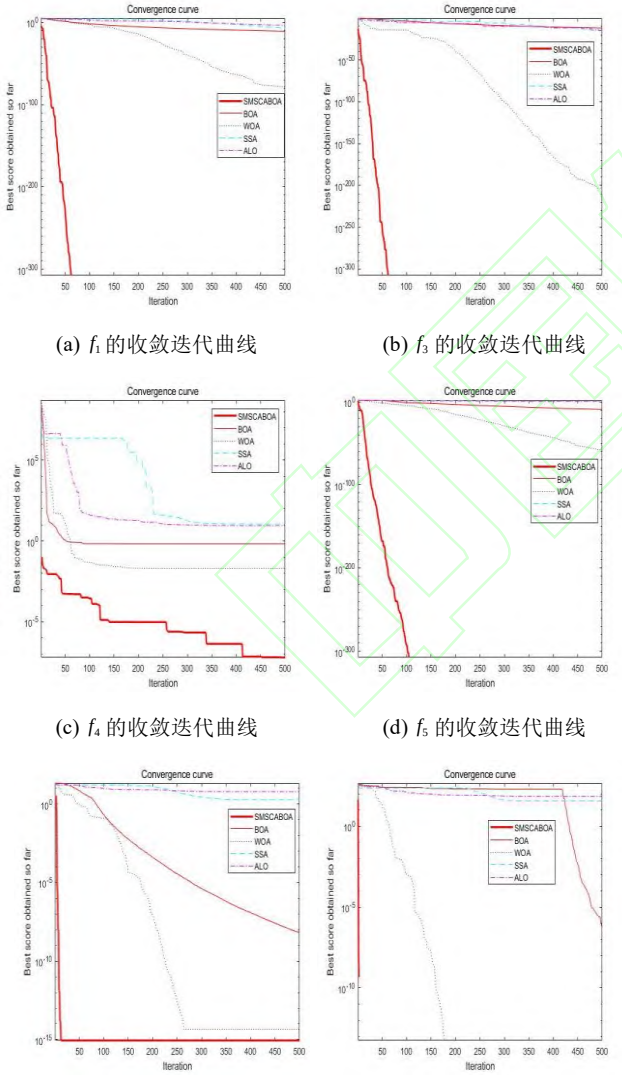
表 3　实验结果

Tab. 3　Experimental result

函数	算法	最优值	平均值	标准差
f_1	SSA	2.58E-08	7.21E-07	1.62E-06
	WOA	8.09E-87	1.15E-73	4.85E-73
	ALO	1.50E-04	1.65E-03	1.36E-03
	BOA	01.08E-11	1.30E-11	1.02E-12
	SCABOA	3.05E-210	2.55E-117	9.55E-117
	SMSCSABOA	**0**	**0**	**0**
f_2	SSA	1.91E-01	4.32E+00	1.33E+01
	WOA	3.13E-57	3.85E-50	1.95E-49
	ALO	1.53E+00	6.19E+01	4.89E+01
	BOA	01.29E-09	4.38E-09	1.36E-09
	SCABOA	7.89E-220	2.35E-119	1.27E-118
	SMSCSABOA	**0**	**0**	**0**
f_3	SSA	1.71E-17	1.35E-15	2.01E-15
	WOA	2.55E-230	3.08E-186	5.63E-192
	ALO	3.47E-17	3.48E-15	4.00E-15
	BOA	4.76E-13	8.89E-13	2.24E-13
	SCABOA	1.64E-228	6.77E-138	3.64E-137
	SMSCSABOA	**0**	**0**	**0**
f_4	SSA	2.85E+00	8.37E+00	3.56E+00
	WOA	5.49E-03	2.68E-02	3.35E-02
	ALO	4.54E+00	1.22E+01	4.58E+00
	BOA	04.14E-01	6.62E-01	1.45E-01
	SCABOA	5.23E-05	4.21E-03	4.99E-03
	SMSCSABOA	**3.88E-09**	**8.20E-07**	**1.048E-06**
f_5	SSA	8.01E-01	3.75E+00	2.05E+00
	WOA	1.05E-58	1.05E-40	5.67E-40
	ALO	2.35E+00	6.61E+00	3.99E+00
	BOA	1.30E-10	6.00E-10	3.58E-10
	SCABOA	1.04E-193	1.57E-159	5.23E-161
	SMSCSABOA	**0**	**0**	**0**
f_6	SSA	1.05E-02	2.63E+00	7.61E-01
	WOA	8.88E-16	4.91E-15	2.55E-15
	ALO	1.16E+00	4.65E+00	2.61E+00
	BOA	4.45E-09	6.05E-09	5.68E-10
	SCABOA	8.88E-16	8.88E-16	**0**
	SMSCSABOA	**8.88E-16**	**8.88E-16**	**0**
f_7	SSA	6.72E-04	1.45E-02	9.80E-03
	WOA	**0**	**0**	**0**
	ALO	2.10E-02	6.84E-02	3.75E-02
	BOA	9.22E-13	4.84E-12	2.22E-12
	SCABOA	**0**	**0**	**0**
	SMSCSABOA	**0**	**0**	**0**
f_8	SSA	1.71E-17	1.35E-15	2.01E-15
	WOA	**0**	**0**	**0**
	ALO	4.18E+01	9.13E+01	2.67E+01
	BOA	1.05E+02	2.80E+01	6.66E+01
	SCABOA	**0**	**0**	**0**
	SMSCSABOA	**0**	**0**	**0**
f_9	SSA	**2.50E-15**	**5.51E-03**	4.81E-03
	WOA	**0**	6.48E-04	2.42E-03
	ALO	**1.38E-13**	5.18E-03	4.85E-03
	BOA	5.32E-12	1.15E-02	7.84E-03
	SCABOA	**0**	**0**	**0**
	SMSCSABOA	**0**	**0**	**0**

为了进一步展示 SMCSABOA 的寻优能力，本文列出了 6 个测试函数的收敛迭代曲线对比图，如图 4 所示，从收敛曲线图可以看出，SMSCABOA 的寻优能力要优于 SSA，WOA，ALO 和 BOA 算法的。

从收敛迭代曲线可以看出，本文的改进算法寻优过程中波动性比较小，鲁棒性好，也间接证实了本文的改进策略能够使 BOA 较快的跳出局部最优。从收敛迭代曲线图上可以看出，基本的 BOA 在 300 代之前迭代的速度是非常缓慢的，说明基本算法易陷入局部最优，且不易跳出局部最优，而改进后 SMSCABOA 在函数 f_1， f_3， f_5， f_8 几乎是呈线性递减的方式收敛到最优值的，收敛速度快，算法收敛到最优解的迭代次数也远远少于基本的 BOA 和其他对比算法，表明将正弦余弦算子作为算法局部搜索时个体更新的新解可以使算法有效的摆脱局部极值的束缚，也间接证实了 limit 阈值和单纯形策略能够使 BOA 避免陷入局部最优，改进算法具有很好的全局勘探性。从函数 f_4 的收敛迭代曲线上可以看出，BOA 算法在收敛到 50 代左右时，基本呈现停滞状态，而改进后的 SMSCABOA 在 50 代后出现多个拐点，也说明改进策略能够使 BOA 有效的跳出局部最优。



(a) f_1 的收敛迭代曲线　　(b) f_3 的收敛迭代曲线

(c) f_4 的收敛迭代曲线　　(d) f_5 的收敛迭代曲线

(e) f_6 的收敛迭代曲线　　(f) f_8 的收敛迭代曲线

图 4　收敛迭代对比图

Fig. 4　Convergent iterative contrast graph

3.5　高维测试函数实验结果及分析

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在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往往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，为了测试改进策略的有效性，进行了 SMSCABOA 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，维数设置为 200 维，其余参数设置设置与 3.2 节相同。实验结果如表 4 所示。

表 4　高维实验结果

Tab. 4　High-dimensional experimental results

函数		SMSCABOA	SMSCABOA	IWOA
		Dim=200	Dim=500	Dim=200
f_1	mean	0	0	2.26E-116
	std	0	0	1.50E-116
f_2	mean	1.54E+200	1.60E+264	1.68E-67
	std	Inf	Inf	1.95E-67
f_3	mean	0	0	--
	std	0	0	--
f_4	mean	1.02E-03	3.83E-03	--
	std	2.02E-03	1.65E-03	--
f_5	mean	0	0	1.66E-69
	std	0	0	2.33E-69
f_6	mean	8.88E-16	8.88E-16	8.88E-16
	std	0	0	0
f_7	mean	0	0	0
	std	0	0	0
f_8	mean	0	0	0
	std	0	0	0

本文在维数为 200 和 500 的条件下的实验，并与文献[20]的对立鲸鱼算法(IWOA)进行对比。从实验结果来看，除函数 f_2 失效之外，其余测试函数均能保证良好的寻优稳定性。函数 f_2 是连续的，平滑的单模态函数，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，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。表中的”--“表示原文献中未进行的函数测试。

4　结束语

本文提出了一种融合正弦余弦以及 limit 阈值以及单纯形法优化的蝴蝶优化算法，结合 limit 阈值使算法有效的退出局部最优解，避免算法早熟，利用单纯形法能够使优化较差蝴蝶的位置，使种群在迭代后期能够较快的收敛到全局最优解；另外本文融合正弦余弦算子，将正弦余弦算法作为局部算子融合到蝴蝶优化算法之中，能够有效的指引蝴蝶个体进行寻优，避免陷入局部最优。本文接下来的研究方向是拓展本文算法在实际工程中的应用

参考文献：

[1] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization [J]. Soft Computing, 2019, 23 (3): 715—734.

[2] Storn R, Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.

[3] Gandomi A H, Yang X S, Alavi A H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems [J]. Engineering with Computers, 2013, 29 (2): 245-245.

[4] Kennedy J, Eberhart R C. Particleswarm optimization [C]// Proceedings

of the IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Service Center, 1995: 1942-1948.

[5] Yang X S. Firefly algorithm, stochastic test functions and design optimization [J]. International Journal of Bio-Inspired Computation, 2010, 2 (2): 78-84.

[6] Salomon R. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms [J]. Biosystems, 1996, 39 (3): 263-278.

[7] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm [J]. 2007, 39 (3): 459-471.

[8] Arora S, Singh S. Butterfly algorithm with Lèvy Flights for global optimization [C]// 2015 International Conference on Signal Processing, Computing and Control (ISPCC) . IEEE, 2015: 220-224.

[9] 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.

[10] Arora S, Singh S. An Improved Butterfly Optimization Algorithm for Global Optimization [J]. Advanced Science, 2017, 8 (9): 711-717.

[11] Arora, S. and Singh, S. “An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization, ”International Journal of Interactive Multimedia and Artificial Intelligence 2017, 4 (4) , 14–21.

[12] 王睿. 植物花授粉算法及应用研究 [D]. 广西民族大学, 2016. (Wang Rui. Plant flower pollination algorithm and its application research [D]. Guangxi University for Nationalities, 2016.)

[13] 易正俊, 韩晓晶. 增强寻优能力的改进人工蜂群算法 [J]. 数据采集与处理, 2013, 28 (6): 761-769. (Yi Zhengjun, Han Xiaojing. Modified artificial bee colony algorithm for search ability enhancement [J]. Journal of Data Acquisition and Processing, 2013, 28 (6): 761-769.)

[14] Nelder J A, Mead R. A Simplex Method for Function Minimization [J]. Commput J, 1965, 7 (4): 308-313.

[15] 郭庆, 惠晓滨, 张贾奎, 李正欣. 多模函数优化的改进花朵授粉算法 [J]. 北京航空航天大学学报, 2018, 44 (04): 828-840. (Guo Qing, Hui Xiaobin, Zhang Jakui, *et al*. Improved flower pollination algorithm for multimodal function optimization [J]. Journal of Beijing University of Aeronautics and Astronautics, 2018, 44 (4): 828-840.)

[16] Mirjalili S. SCA: A Sine Cosine Algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2016, 96 (2016): 120-133.

[17] Mirjalili S, Gandomi A H, Mirjalili S Z, *et al.* Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems [J]. Advances in Engineering Software, 2017: S0965997816307736.

[18] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[19] Mirjalili S. The ant lion optimizer [J]. Advances in Engineering Software, 2015, 83: 80-98.

[20] 龙文, 蔡绍洪, 焦建军, 唐明珠, 伍铁斌. 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. (Long Wen, Cai Shaohong, Jiao Jianjun, *et al.* Improved whale optimization algorithm for large scale optimization problems [J]. Systems Engineering-Theory &Practice, 2017, 37 (11): 2983-2994.)