

基于非线性云化的自适应帝王蝶优化算法

李晓平¹, 杜波^{1†}, 王贤文²

(1. 长安大学人文学院, 陕西 西安 710054; 2. 大连理工大学人文与社会科学学部, 辽宁 大连 116024)

摘要: 针对帝王蝶算法多样性退化、易陷入局部最优导致寻优精度不高的问题, 提出一种基于非线性云化的自适应帝王蝶算法(NCSMBO). 深入探究帝王蝶算法的进化机制, 指出其本质为网格搜索算法; 在迁移和调整算子中, 采用正向正态云发生器对父代帝王蝶个体执行非线性云化操作, 增加候选解的数量, 提高局部开发能力; 对云化后的后代个体引入贪婪策略, 增强算法的可行性; 为从发生概率上对突变进行控制, 进一步给出双圆正切形式的自适应调整率. 在 12 个不同特征基准测试函数上对包含 NCSMBO 在内的 7 种优化算法进行综合评估以及对两类数学规划问题求解验证, 实验结果均表明所提算法具有更高的收敛精度和稳定性.

关键词: 帝王蝶算法; 网格搜索; 非线性云化; 贪婪策略; 自适应调整率; 收敛精度

中图分类号: TP301.6 文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0448

引用格式: 李晓平, 杜波, 王贤文. 基于非线性云化的自适应帝王蝶优化算法[J]. 控制与决策.

Self-adaptive monarch butterfly optimization based on nonlinear cloud-transfer

LI Xiao-ping¹, DU Bo^{1†}, WANG Xian-wen²

(1. School of Humanities and Social Science, Chang'an University, Xi'an 710054, China; 2. Faculty of Humanities and Social Science, Dalian University of Technology, Dalian 116024, China)

Abstract: In order to solve the problem of low precision of monarch butterfly optimization(MBO)causing by diversity degradation and being easy to fall into local optimum, a self-adaptive monarch butterfly optimization based on nonlinear cloud-transfer(NCSMBO) is proposed. Evolution mechanism of MBO is deeply explored which indicates the nature of MBO is grid search. Nonlinear cloud-transfer operation to parent monarchs is executed utilizing a forward normal-cloud generator in both migration operator and adjusting operator, which can increase candidate solutions and improve the ability of local exploitation. Meanwhile, a greedy strategy is introduced in offspring from cloud-transfer to enhance the feasibility. A self-adaptive adjustment rate in the form of double-circle-tangent is given to control mutation based on occurrence probability. Seven optimization algorithms including NCSMBO are overall evaluated on 12 benchmark functions with different feature, and two types of mathematical planning problem are solved and verified. All of the simulation results show that the proposed algorithm has better convergence accuracy and stability.

Keywords: monarch butterfly optimization algorithm; grid search; nonlinear cloud-transfer; greedy strategy; self-adaptive adjustment rate; convergence accuracy

0 引言

受生物体社会行为启发的群智能进化算法具有机制简单、可操作性强、无需梯度计算等优点, 在解决大规模非线性复杂问题时表现出了良好效果, 已成为当前人工智能领域的一个研究热点, 如经典的遗传算法(Genetic Algorithm, GA)^[1]、粒子群优化(Particle Swarm Optimization, PSO)^[2]以及近年来出现的狼群算法(Wolf Pack Algorithm, WPA)^[3]、鲸鱼优

化算法(Whale Optimization Algorithm, WOA)^[4]、人工蜂群算法(Artificial Bee Colony, ABC)^[5]以及帝王蝶优化算法(Monarch Butterfly Optimization, MBO)^[6]等, 其中 MBO 以原理简单、调整参数少、便于并行化处理等方式被给予广泛应用, 相较于其它进化算法具有一定的竞争力. 为应对不同领域需要, 当前也相继发展出了多目标 MBO^[7]、离散 MBO^[8]、二进制 MBO^[9-10]及混合 MBO^[11-12]等案例形式, 均取得了

收稿日期: 2022-03-21; 录用日期: 2022-07-06.

基金项目: 国家自然科学基金项目(71974029); 中央高校基本科研业务费专项经费资助项目(300102112608); 陕西省社会科学基金项目(2021R013).

†通讯作者. E-mail:tbca_0929@126.com.

良好效果.

MBO 的灵感源于北美洲帝王蝶群的季节迁徙行为, 它将整个种群划分为两个子种群, 相互之间采取协同进化机制, 通过合作与竞争不断地交换信息, 促使整个种群动态演化, 同时利用迁移率、调整率等参数控制搜索方向, 进而找到最优解. 与其它进化算法类似, MBO 同样存在收敛精度不高、易陷入局部最优以及求解不稳定等问题, 为改善优化性能, 研究者们从诸多方面对 MBO 进行了改进.

在国内方面, 冯艳红等^[13]从记忆种群最优解历史经验的角度出发, 基于差分进化思想利用差分变异对迁移算子进行改进, 实现了种群内部信息的充分共享. 孙林等^[14]也利用差分变异与迁移算子融合, 保证新迁移算子具有更强的探索能力, 同时在每次迭代末段追加较差帝王蝶个体的柯西变异过程, 强化了种群的多样性. 在国外方面, Hossam 等^[15]为提高搜索精度和效率, 取消 levy 飞行策略, 代之以开关量阈值辅助改进调整算子, 通过比较开关量与阈值来决定是否使用前一代的全局最优信息. Sun 等^[16]提出了一种基于反向学习和随机局部扰动的帝王蝶优化算法, 结合反向学习策略丰富种群初始化的多样性, 采用随机局部扰动避免陷入局部最优. Abdelmonem 等^[17]为充分利用种群共享信息, 在迁移和调整算子中将个体进化方式修正为基于维度方向的均值而非单个个体, 促进了算法快速收敛; 同时, 除原有调整率外在两种算子中又设计了一种软调整率, 在为其设置极小值的条件下有助于减少陷入局部最优的概率. Kaviarasan 等^[18]为解决云环境下负载不均衡问题, 在响应时间差基础上为 MBO 定义了响应阈值的概念, 以此为基础为迁移算子做了简化和调整, 结果表明所提改进算法有较好的适用性. Jyotirmayee 等^[19]将正余弦算法思想引入 MBO 中, 充分借助正余弦算法的强全局探索能力, 在迁移算子和调整算子中分别增加随机正弦或余弦偏移量, 以规避局部最优, 提升 MBO 全局开发能力.

以上改进策略虽然均取得了一定的性能提升, 但未从进化本质上进行探究, 无法较好地平衡收敛速度与搜索精度的关系, 寻优效率有待进一步增强. 经分析发现, 算法在逼近全局最优解过程中, 实际利用 Levy 飞行策略突破“早熟”. 若理论最优解不位于种群进化的历史轨迹上, 仅依赖 Levy 飞行策略突破原取值空间的限制, 不仅概率偏低, 而且难以逼近理论最优解. 该现象可归纳为种群多样性低, 甚至会出现多样性的减少, 导致局部搜索不充分, 另

外恒定参数也容易产生探索与开发能力之间不均衡的问题.

针对以上问题, 本文提出一种基于非线性云化的自适应帝王蝶算法 (Self-Adaptive Monarch Butterfly Optimization Based on Nonlinear Cloud-transfer, NCSMBO). 具体为: 首先, 考虑到云模型的不确定性转化能力更适用于前后代之间的进化机制, 对目标父代个体进行云化, 凸显搜索的充分性, 云化程度也随着迭代进行而降低, 保证算法的收敛性; 其次, 基于贪婪策略, 在云化之后选择较优个体; 最后, 设计自适应变化的调整率, 随迭代进行变异调整的程度变低. 在 12 个基准测试函数上的对比实验表明, 改进算法具有更为出色的性能.

1 帝王蝶优化算法 (MBO)

假设帝王蝶个体为 $x_i \in R^d$, 其中 R^d 为 d 维解空间, $i = 1, 2, \dots, N$, N 为种群规模. MBO 将整个帝王蝶种群划分为 2 个子种群 SP_1 和 SP_2 , 其包含的个体数量分别为 N_1 和 N_2 . 假设 SP_1 占比为 p (也称迁移率), 则有:

$$\begin{cases} N_1 = \text{ceil}(Np); \\ N_2 = N - N_1. \end{cases} \quad (1)$$

其中函数 $\text{ceil}(\cdot)$ 代表向上取整. MBO 的种群进化主要基于两种算子: 迁移算子 (Migration Operator, MO) 和调整算子 (Adjusting Operator, AO).

迁移算子负责实现子种群 SP_1 内部及与子种群 SP_2 间的单向信息交换, 其数学模型可描述为:

$$x_{i,k}^{t+1} = \begin{cases} x_{r1,k}^t, & \text{if } r \leq p; \\ x_{r2,k}^t, & \text{else.} \end{cases} \quad (2)$$

其中, $x_{i,k}^{t+1}$ 表示个体 x_i 迭代至第 $t+1$ 代时的 k 维分量, 且 $x_i \in SP_1$; x_{r1} 、 x_{r2} 分别为从 SP_1 和 SP_2 中随机选择的两个子个体; 变量 $r = \text{rand} \cdot \text{peri}$, rand 为 $[0,1]$ 上服从均匀分布的随机数; peri 为迁移周期.

调整算子利用社会效应对子种群 SP_2 进行位置更新, 促进算法的快速收敛, 其更新方式如下:

$$x_{i,k}^{t+1} = \begin{cases} x_{best,k}^t, & \text{if } \text{rand} \leq p; \\ x_{r3,k}^t, & \text{else.} \end{cases} \quad (3)$$

其中, $x_i \in SP_2$; x_{best} 表示当前发现的最优个体; x_{r3} 为来自 SP_2 中的随机个体.

进一步进行变异操作. 若 $\text{rand} > \text{BAR}$, 则按以下方式继续更新 $x_{i,k}^{t+1}$:

$$x_{i,k}^{t+1} = x_{i,k}^{t+1} + \alpha(dx_k - 0.5). \quad (4)$$

其中, BAR 为调整率; α 为权重因子, 利用 $\alpha =$

S_{\max}/t^2 得出, 其中 S_{\max} 为可移动的最大调整步长, t 为当前的迭代步数; dx_k 表示帝王蝶个体的步长, 以 Levy 函数计算得出:

$$dx = Levy(x_i^t). \quad (5)$$

2 基于非线性云化的自适应帝王蝶优化算法 (NCSMBO)

2.1 MBO 进化机制分析

从算法机理上看, MBO 采用的是类似离散数值空间的策略, 执行迭代前整个种群的所有维度会组成一个随机多维取值网格, 其度量了种群的多样性. 整个迁移算子以及到调整算子变异之前的步骤覆盖了 MBO 的主要部分, 在该部分中, 每一代个体的更新采用网格化跳跃的方式, 即后代个体的产生立足于当前种群的网格布局, 其每一维上的信息完全继承自随机选择的父代个体, 可将此形象化为网格化迁移. 显然, MBO 的这种进化机制隐含着社会效应, 然而却无法获得新的进化信息, 后代个体的繁衍主要来自父代个体各维度信息的随机交叉重组. 下面以二维单峰函数 $f(X, Y) = X^2 + Y^2$ 求解最小值为例对迁移算子进行说明 (见示意图 1).

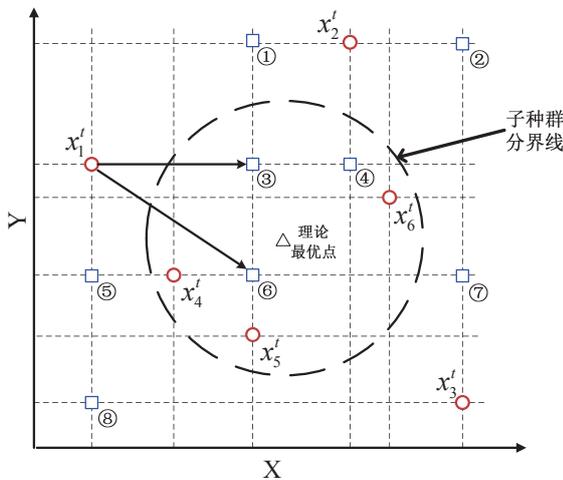


图 1 迁移算子的进化

图 1 中, \circ 表示第 t 代的帝王蝶个体, \square 表示其 $t + 1$ 代新产生的后代. 假设 $N = 6$, 所有个体的位置分布构成了二维取值网格, 子种群 $SP_1 = \{x_1^t, x_2^t, x_3^t\}$ 与 $SP_2 = \{x_4^t, x_5^t, x_6^t\}$ 由圆形虚线分割. 以个体 x_1^t 为例, 根据迁移算子, 共存在单维或二维方向上同时进化两种情形. 显然, 其后代个体共可产生 $N^d=36$ 种结果, 图 1 中序号 ①-⑧ 为其代表性的分布位置. 例如满足 $(r_2 = 5|k = 1, r > p) \& (r_1 = 1|k = 2, r \leq p)$ 条件下随机产生个体 ③, 在 $(r_2 = 5|k = 1, r > p) \& (r_1 = 1|k = 2, r > p)$ 条件下随机产生个体 ⑥. 可以发现, 新的后代个体不论如何

分布, 它们均位于网格线的交点上, 各维取值均来源于上一代, 这是由算法特性决定的, 无法摆脱网格限制. 调整算子的进化规律也类似.

打破以上困境的唯一手段为变异操作. 由于式 (4) 中 $\alpha(dx_k - 0.5)$ 为连续型随机数, 个体会以一定变异概率在 X 维上随机产生新刻度, 且与原取值空间大概率无交集, 因此在后续迭代时会成为新取值来源. 在增加种群规模、提高突变概率的条件下, 容易产生更多新刻度, 算法也就有更多机会跳出局部最优. 然而, 种群规模通常设置为固定值; 提高突变概率在进化前期可保持较高的种群多样性, 但在后期会严重影响收敛速度.

经上述分析可知: (1)MBO 利用 Levy 飞行策略增加网格分布密度, 然而新网格是随机的, 不一定分布在最优解附近, 因此需要更多迭代和更高的调整率才能高精度搜索, 自身认知的局部开发能力难以保证.(2) 调整率决定了突变的概率. 由于其为固定值, 过小取值虽然会在前期增加突变概率和取值多样性, 但后期难以保证收敛速度, 且容易跳出最优点; 反之, 过大取值会限制前期突变, 算法容易快速收敛, 较早地陷入局部最优.

2.2 改进算法 NCSMBO

针对以上问题, 为实现精细搜索, 提高解的精度, 提出非线性云化设计策略, 在迁移和调整算子中对父代个体进行云化; 同时, 为平衡全局探索和局部开发能力, 采取自适应调整的方式, 引入非线性变化的调整率. 下面以最小化问题为例进行阐述.

2.2.1 非线性云化设计

云模型^[20] 是一种应用于定性概念与定量分析转换的不确定性分析模型, 它常用一个三元组 (Ex, En, He) 表示整体特性, 其中期望 Ex 、熵 En 、超熵 He 分别为描述云模型的 3 个数字特征. 由给定数字特征生成云滴的映射过程称为正向云发生器^[21], 可实现由定性概念到数值空间之间的不确定性转换.

针对第一个缺陷, 可基于云模型通过丰富取值空间的方式解决. 具体思路为: 不论对于迁移算子或调整算子, 在后代个体上, 对目标父代个体均执行云化, 以破除原取值网格的限制, 增加进化的不确定性, 同时大概率保留原父代个体的位置信息, 保留之前的进化成果. 由于正态分布的普适性, 此处采用正向正态云发生器^[22] 实现以上云化操作, 这也是应用范围最广的一种云模型, 其形式如下:

$$[x_1, x_2, \dots, x_n] = G(Ex, En, He, n). \quad (6)$$

其中 n 表示生成的云滴数, 正向正态云发生器的实

现过程可参考文献 [22].

对于迁移算子, 利用正向正态云发生器对随机选择的两父代个体 x_{r1} 、 x_{r2} 分别执行云化操作, 以生成新的取值空间信息. 作为云的重心, Ex 的设置不改变初始策略, 仍以随机个体的位置为基准, 以体现保守能力; En 代表探索范围, 不同的 En 衡量后代个体与 x_{r1} 或 x_{r2} 之间的径向距离; 调整 He 可获得不同的个体离散度, 过大取值会丧失稳定性, 过小又会降低随机性; 云滴数 n 决定了生成后代的数量. 据此给出新的迁移算子更新公式:

$$x_{i,k}^{t+1} = \begin{cases} G(x_{r1,k}^t, En, He, n), & \text{if } r \leq p; \\ G(x_{r2,k}^t, En, He, n), & \text{else.} \end{cases} \quad (7)$$

$$\begin{cases} En = \frac{\delta(X_{\max}(k) - X_{\min}(k))}{\tan(\pi t / (4T_{\max}) + \pi/4)}; \\ He = En/10. \end{cases} \quad (8)$$

其中, δ 为控制系数, 用于调节云滴的不确定性分布; $X_{\max}(k)$ 、 $X_{\min}(k)$ 分别为第 k 维的边界最大值和最小值; T_{\max} 为最大迭代次数. 可以发现, En 和 He 采取了非线性递减的形式, 前期需体现个体探索的强不确定性, 对 En 和 He 取较大值; 后期为防止云滴过度发散, 对 En 和 He 进行限制.

新的迁移算子仍然基于维度层面实现进化, 而非以个体向量的形式, 因此对所有维度云化后会获得 n^d 个后代, 此时种群规模呈现幂级增长, 不仅与算法设计思想相悖, 而且会极大增加算法复杂度和计算开销, 难有可行性. 针对此问题, 本文引入贪婪策略, 即利用当前父代个体 x_{r1} 或 x_{r2} 衡量新信息产生的优劣, 从中选择更优的解空间. 对于最小化问题, 当 $r \leq p$ 时, 选择过程可表示为:

$$x_{i,k}^{t+1} = \begin{cases} x_{i,k}^{t+1}, & \text{if } \min(f_j(x_i^{t+1})) < f(x_{r1}), \\ j = 1, 2, \dots, n; \\ x_{r1,k}, & \text{else.} \end{cases} \quad (9)$$

当 $r > p$ 时:

$$x_{i,k}^{t+1} = \begin{cases} x_{i,k}^{t+1}, & \text{if } \min(f_j(x_i^{t+1})) < f(x_{r2}), \\ j = 1, 2, \dots, n; \\ x_{r2,k}, & \text{else.} \end{cases} \quad (10)$$

其中, $f_j(x_i^{t+1})$ 表示与第 j 个云滴相对应的新个体 x_i^{t+1} 的适应度. 如果 n 个云滴中的最优云滴适应度小于 x_{r1} , 则视之为较优云滴, 将其取代 x_{r1} , 实现信息更新; 否则全部视为无效云滴, 后代个体的取

值仍然按照式 (2) 执行. 如此设计既能满足种群规模不变的设置, 又能使后代继承方式更兼多元化, 通过引入更优的探索信息, 增加候选解的数量, 保证解空间进一步向最优解靠近.

对于调整算子, 若 $rand \leq p$, 则对当前的最优个体不执行云化操作, 便于保持当前最优解. 其原因在于, 最优个体代表了当前搜索的最优成果, 云化之后虽然也会增加解空间的多样性, 但之后产生的最优解可能会发生偏离, 抵消前期搜索的成果, 降低收敛速度. 反之, 若 $r > p$, 则采取与迁移算子相仿的改进方式. 具体实现如下:

$$x_{i,k}^{t+1} = \begin{cases} x_{best,k}^t, & \text{if } rand \leq p; \\ G(x_{r3,k}^t, En, He, n), & \text{else.} \end{cases} \quad (11)$$

其中 En 和 He 的取值方式与式 (8) 相同. 若存在云化操作, 则最优云滴的选择仍采取贪婪策略, 不再赘述. 另外, 因突变过程得到的是随机值, 几乎不存在于原解空间中, 本身就能增加种群多样性, 故不需要进行云化.

2.2.2 自适应调整率设计

调整率 BAR 体现的是帝王蝶个体突变的能力. 在进化前期, 若个体搜索步长较大, 则可在更广范围内探索, 较快地定位至全局最优解附近, 也能避免较早陷入局部极值, 此时 BAR 应取较小值, 留给突变更大的概率区间; 进化后期为缩小搜索范围, 能深入地在局部区域挖掘, 应降低突变率, 保持较大的 BAR 值. 为达到这一目标, 克服 BAR 固定取值的缺点, 提出如下自适应调整率:

$$BAR = 1 - \frac{\exp(2(1 - t/T_{\max})) - \exp(-2(1 - t/T_{\max}))}{\exp(2(1 - t/T_{\max})) + \exp(-2(1 - t/T_{\max}))}. \quad (12)$$

其满足 $0 \leq BAR < 1$. 可以看到, BAR 的计算采用了双圆正切的形式, 趋势变化呈现出前慢后快的规律, 随着迭代进行能够实现对其自适应调整.

2.2.3 改进算法的流程

结合上述改进, 给出 NCSMBO 的实现流程, 具体见算法 1.

3 仿真实验与分析

3.1 基准测试函数与评价标准

为公平起见, 选取 12 个 Benchmark 函数作为基准测试函数^[23](见表 1), 它们均属于求解最小值问题, 理论最优值均为 0. 各函数具有以下不同特征: 单峰 (Unimodal)、多峰 (Multimodal)、可分 (Separable)

算法 1 NCSMBO

Input: $N, d, p, T_{\max}, S_{\max}, peri$, control coefficient δ , cloud drop number n ;

Output: global optimal solution x^* ;

```

1: Initialize the population and compute corresponding individual fitness;
2: while  $t \leq T_{\max}$  do
3:   Sort the population by individual fitness;
4:   Divide the whole population into two subpopulation ( $SP_1$  and  $SP_2$ ) based on Eq.(1);
5:   for  $i = 1 : N_1$  do
6:     for  $i = 1 : d$  do
7:       Generate  $n$  offspring individuals for  $x_{i,k}^t$  based on forward normal cloud generator and Eq.(7)~(8);
8:       Randomly generate  $r$ ;
9:       Update  $x_{i,k}^{t+1}$  by selecting the better cloud drop based on Eq.(9)~(10) by comparing  $r$  with  $p$ ;
10:    end for
11:  end for
12:  for  $i = 1 : N_2$  do
13:    for  $i = 1 : d$  do
14:      Randomly generate  $r$ ;
15:      if  $r > p$  then
16:        Generate  $n$  offspring individuals for  $x_{i,k}^t$  based on the cloud generator, Eq.(8) and Eq.(11);
17:        Update  $x_{i,k}^{t+1}$  by selecting the better cloud drop based on Eq.(9)~(10);
18:      end if
19:    end for
20:  end for
21:  Update  $BAR$  based on Eq.(12) and further perform the variation based on Eq.(4)~(5);
22:  Combine  $SP_1$  with  $SP_2$  into a whole population and compute its fitness;
23: end while

```

及不可分 (Non-Separable), 分别简化为 U/M/S/N. 为保持公平性, 所有实验独立进行 30 次, 最大迭代次数为 1000, 种群规模为 30, 变量维度取 10.

表 1 基准测试函数

函数	函数名	取值范围	特征
f_1	Step	[-100, 100]	U/S
f_2	Sphere	[-5.12, 5.12]	U/S
f_3	Quartic	[-1.28, 1.28]	U/S
f_4	Schwefel 2.21	[-100, 100]	U/S
f_5	Schwefel 2.22	[-10, 10]	U/N
f_6	Rosenbrock	[-30, 30]	U/N
f_7	Rastrigin	[-5.12, 5.12]	M/S
f_8	Schwefel 2.26	[-500, 500]	M/S
f_9	Ackley	[-35, 35]	M/N
f_{10}	Griewank	[-100, 100]	M/N
f_{11}	Salomon	[-100, 100]	M/N
f_{12}	Zakharov	[-5, 5]	M/N

为有效衡量不同算法间的性能差异, 共选取最优值、平均值、标准差、求解成功率 (Solve Success Rate, SSR)、耗时 (Eclipse time, ET) 等 5 项指标. 求解成功率定义为满足成功条件的测试次数与测试总次数的比值, 按下式判断每次求解是否成功:

$$|f_{\text{best}} - f_0| < 10^{-5}. \quad (13)$$

其中, f_{best} 为每次求解获得的最优值; f_0 为理论最

优值.

3.2 关键参数的性能实验及分析

NCSMBO 中额外引入了两个关键参数: 云滴数 n 和控制系数 δ . 本文拟根据正交试验的思想, 通过改变各参数值来检验算法的收敛效果. 对于 n , 将其从 1 逐渐增至 20, 兼顾时间复杂度的考虑, 在 $n > 5$ 后仅选取了几个代表性的点; 对于 δ , 从 0.1 开始, 以步长 0.1 逐渐增加至 1.0; 在每种参数组合下对测试函数独立重复 30 次实验, 然后求取平均值作为综合收敛精度; 算法涉及的其它参数均取自原始文献 [6] 的设定: $p = 5/12$, $peri = 1.2$, $S_{\max} = 1$. 限于篇幅, 下面仅列出函数 f_7 的实验结果 (见表 2).

对比两参数遍历的平均最优解可知: 不论参数取值如何组合, 函数的平均最优解均满足求解成功的条件, 最优解精度可达 10^{-18} 数量级. 当 n 固定时, 随着控制系数 δ 的增大, 函数的求解结果逐渐变差, 在取 0.1 时可获得最优的综合收敛精度, 表明此时云滴的不确定性分布得以充分协调, 能取得最佳的精度收益. 为此, 本文将 δ 的最优取值设置为 0.1.

另外也可发现, 随着 n 的增加, 函数的平均最优解整体呈现出减小趋势, 收敛精度也随之提升. 这是由于云滴数度量了算法的探索范围和力度, 更多的云滴象征着更广的取值空间. 在耗时方面, 当 $n \leq 5$

表2 不同参数组合下测试函数 f_7 的平均最优解

δ	n								
	1	2	3	4	5	10	15	20	
0.1	4.43e-17	3.72e-17	2.89e-17	1.90e-17	1.15e-17	3.88e-18	3.80e-18	2.56e-18	
0.2	1.42e-16	9.77e-17	9.82e-17	9.77e-17	8.80e-17	5.22e-17	4.86e-17	4.50e-17	
0.3	2.80e-16	1.23e-16	1.13e-16	9.30e-17	8.86e-17	5.82e-17	5.17e-17	5.02e-17	
0.4	5.01e-16	1.65e-16	1.68e-16	1.54e-16	1.35e-16	1.69e-16	4.44e-16	4.14e-16	
0.5	2.23e-14	9.93e-15	9.89e-15	8.23e-15	7.99e-15	6.20e-15	5.13e-15	3.87e-15	
0.6	7.46e-14	4.66e-14	4.24e-14	3.53e-14	3.06e-14	2.50e-14	1.31e-14	1.02e-14	
0.7	7.80e-13	6.21e-13	6.00e-13	5.57e-13	4.39e-13	3.19e-13	8.88e-14	6.62e-14	
0.8	8.33e-13	7.52e-13	7.21e-13	6.84e-13	6.20e-13	4.40e-13	3.36e-13	3.11e-13	
0.9	8.86e-13	7.60e-13	7.02e-13	6.99e-13	6.50e-13	4.84e-13	4.23e-13	3.80e-13	
1.0	6.02e-12	3.55e-12	2.54e-12	2.13e-12	1.84e-12	6.61e-13	5.55e-13	5.23e-13	

时算法总耗时较少,继续增大 n 则会带来额外的计算开销.由于 NCSMBO 求解不同函数的区别主要在于公式上,因此同一云滴数下的耗时较为相近.随着 n 的增加,算法带来的精度增益逐渐降低,在 n 较小时收敛精度提升已十分明显,继续增加 n 后性能提升十分有限,有必要在满足精度要求下尽量选择较小 n 值.深入探究发现,在 $n = 2$ 时收敛精度的变化拐点较为明显,因此为达到收敛精度与耗时之间的良好平衡,实现效益最大化,本文将 n 设置为 2.

3.3 不同改进策略的有效性分析

NCSMBO 在 MBO 基础上分别引入了非线性云化策略和自适应调整率策略,为检验不同策略对算法性能的影响,分别将 NCSMBO 与仅采用非线性云化策略的算法 (NCMBO)、仅采用自适应调整率策略的算法 (SAMBO) 进行对比,具体结果见表 3.

由结果可知: NCSMBO 整体上取得了最好的最优值、平均值及标准差,表明同时融合非线性云化与自适应调整率策略的帝王蝶算法可有效提升原算法的全局寻优能力;相比之下,NCMBO 的性能表现仅次于 NCSMBO, SAMBO 整体寻优效果最差.在求解成功率方面, NCSMBO 表现出了更强的性能优势,在 9 个函数上以 100% 的成功率发现了符合条件的最优解.在耗时方面, NCSMBO 和 NCMBO 均引入了非线性云化策略,带来了较高的运算开销,耗时也较长; SAMBO 相对于 MBO 仅增加了式 (12) 的简单计算,耗时更短,但在收敛精度上明显劣于其它两种对比算法.

综上所述,非线性云化策略在 NCSMBO 性能提升方面占主导作用,自适应调整率策略仅影响帝王蝶个体在进化不同阶段的突变能力,性能帮助效果有限;两种策略在融合后共同作用,全局与局部搜索能力可实现较好平衡,大大提升算法的收敛精度.

3.4 不同进化算法间的性能对比实验

为验证改进帝王蝶算法的有效性,分别将本文 NCSMBO 算法、差分进化帝王蝶算法 (DEMBO)^[13]、基于反向学习和随机局部扰动的帝王蝶算法 (OPMBO)^[16]、基于开关量阈值的改进帝王蝶算法 (STMBO)^[15]、基本帝王蝶算法 (MBO)^[6]、传统粒子群优化算法 (PSO)^[2] 以及较新颖的天牛须搜索算法 (Beetle Antennae Search, BAS)^[24] 进行对比,所有算法的参数取值均源于相应出处文献的建议值;变量维度分别取为 10 和 100.因实验结果篇幅较大,本文将具有统计特征的箱形图 (以 $d = 10$ 为例) 以及各实验指标的数值结果放于附录中.

由箱形图可发现,在区间分布上, NCSMBO 整体上呈现出最优的求解结果;从 25% 和 75% 分位数、中值数以及异常值的角度对比优势更为明显,充分说明 NCSMBO 可对大多数函数实现全局最佳寻优,收敛精度和计算稳定性表现更好,结果分布更紧凑.

除此之外,根据数值结果可知:当 $d = 10$ 时,包含 NCSMBO 在内的帝王蝶算法具有较好的寻优能力,相比其它算法成功率更高,大多数函数上可达到 100%.在计算耗时上, BAS 最少,帝王蝶类算法最多.究其原因, BAS 仅依靠单个天牛个体寻找食物源,进化速度最快;所有帝王蝶算法的进化机制因涉及到维度层面计算,计算开销也稍多,而 NCSMBO 因具有迁移算子和调整算子的云化、调整率自适应变化等过程,能够更精细地搜索更多解,相较于 MBO 耗时也有所增加,但与其它改进帝王蝶算法的耗时十分相近.当维度提高至 100 时,问题复杂度也呈现指数级增长,求解难度递增,导致函数的收敛精度伴随着同步下降,而 NCSMBO 受此影响变化相对较小,寻优精度也相对最高, PSO 和 BAS 的收敛精度最差.例如在函数 f_{12} 上仍能求得 10^{-11} 数量级的最优值,远优于其它对比算法.

表 3 三种改进策略的寻优结果对比

函数	指标	NCSMBO	NCMBO	SAMBO	函数	指标	NCSMBO	NCMBO	SAMBO
f_1	最优值	4.05e-22	4.77e-22	6.20e-22	f_7	最优值	4.59e-19	1.30e-17	2.08e-01
	平均值	4.57e-22	5.75e-22	9.16e-22		平均值	3.72e-17	8.00e-17	8.84e-01
	标准差	3.31e-22	3.33e-22	4.00e-22		标准差	1.85e-16	2.32e-16	1.99e+00
	SSR(%)	100	100	100		SSR(%)	100	100	0
	ET(s)	2.37	2.30	2.13		ET(s)	3.47	3.42	3.16
f_2	最优值	1.06e-24	5.30e-24	1.67e-21	f_8	最优值	1.01e-08	5.54e+00	3.26e+02
	平均值	1.26e-24	6.80e-22	1.86e-21		平均值	2.50e+03	2.26e+03	2.44e+03
	标准差	1.21e-23	2.99e-19	7.10e-20		标准差	5.27e+02	1.26e+02	3.33e+02
	SSR(%)	100	100	100		SSR(%)	7	0	0
	ET(s)	3.13	3.06	2.90		ET(s)	4.27	4.26	3.94
f_3	最优值	1.05e-08	1.14e-07	1.20e-06	f_9	最优值	1.07e-09	3.64e-09	4.12e-07
	平均值	5.51e-06	6.46e-06	6.02e-06		平均值	3.04e-09	4.44e-09	4.98e-07
	标准差	8.76e-06	8.24e-06	8.80e-04		标准差	8.16e-08	1.02e-08	1.03e-07
	SSR(%)	100	100	100		SSR(%)	100	100	100
	ET(s)	3.39	3.41	2.65		ET(s)	3.60	3.58	3.37
f_4	最优值	4.22e-12	4.50e-12	4.67e-12	f_{10}	最优值	1.55e-02	2.10e-02	3.07e-02
	平均值	4.57e-12	4.87e-12	3.32e-11		平均值	3.10e-02	4.48e-02	5.55e-01
	标准差	5.46e-11	5.80e-11	7.71e-11		标准差	1.20e-02	1.30e-01	7.21e-01
	SSR(%)	100	100	100		SSR(%)	0	0	0
	ET(s)	3.15	3.07	2.80		ET(s)	3.56	3.51	3.31
f_5	最优值	1.47e-16	2.77e-15	6.20e-11	f_{11}	最优值	8.12e-07	6.31e-07	7.07e-07
	平均值	6.46e-16	6.64e-14	4.08e-09		平均值	1.41e-06	1.95e-06	2.19e-06
	标准差	1.13e-16	2.22e-13	7.00e-09		标准差	1.14e-06	1.10e-06	4.40e-06
	SSR(%)	100	100	100		SSR(%)	100	100	100
	ET(s)	3.31	3.27	2.93		ET(s)	4.19	4.10	3.62
f_6	最优值	4.11e-03	6.65e-03	5.10e-02	f_{12}	最优值	4.56e-13	6.60e-12	5.24e+08
	平均值	1.21e-02	3.24e-02	3.22e+01		平均值	5.50e-13	1.54e-11	8.08e+08
	标准差	8.35e-02	7.20e-02	1.13e+01		标准差	9.11e-13	7.74e-11	5.24e+08
	SSR(%)	0	0	0		SSR(%)	100	100	0
	ET(s)	4.24	4.20	3.78		ET(s)	3.71	3.68	3.15

3.5 收敛曲线性能分析

收敛曲线作为衡量算法性能的重要指标，可以很好地反映算法陷入局部最优解的时间和次数。图 2~5 给出了当维度取 10、迭代次数设置为 2000 时部分函数的平均收敛曲线；为方便观察各算法的收敛性能，将纵坐标设置为以 10 为底的适应度对数值。

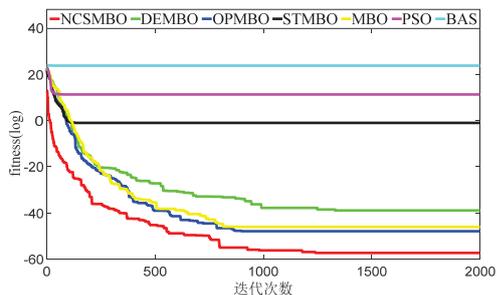


图 2 函数 f_2 的平均收敛曲线

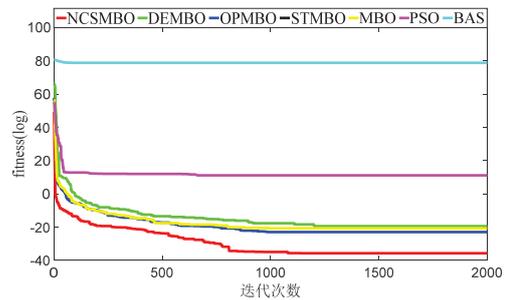


图 3 函数 f_5 的平均收敛曲线

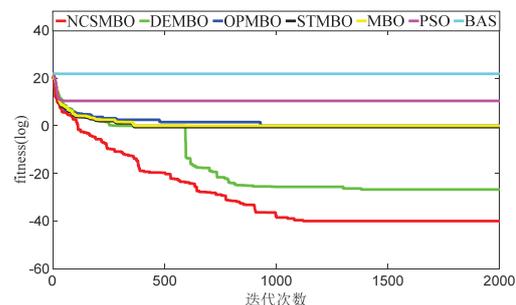


图 4 函数 f_7 的平均收敛曲线

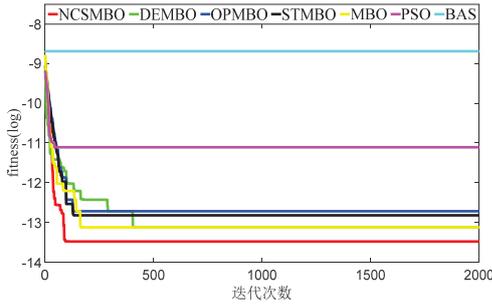


图 5 函数 f_{11} 的平均收敛曲线

观察各函数的平均收敛曲线可知, 各对比算法在迭代后期乃至结束后收敛曲线基本都能实现平稳. 但相比之下, NCSMBO 能以更快的速度进化, 最终收敛精度也最高; 其它对比算法在大部分函数上过早陷入局部极值, 难以跳出束缚, 导致迭代结束时收敛精度也相对较低.

3.6 数学规划问题的求解实验及对比分析

数学规划问题指的是在特定约束条件下求解某目标函数的极值问题, 应用范围广泛. 为检验本文 NCSMBO 算法对数学规划类问题的适用性, 分别设计如下线性规划 (LP) 与非线性规划 (NLP) 问题:

$$\begin{aligned}
 & \text{LP: } \min 3x_1 + 4x_2 + 6x_3. \\
 & \text{s.t. } \begin{cases} x_1 - x_2 + x_3 \leq 20; \\ 3x_1 + 2x_2 + 4x_3 \leq 42; \\ 3x_1 + 2x_2 \leq 30; \\ x_1 + x_2 - x_3 = 7; \\ x_1, x_2, x_3 \geq 0. \end{cases} \quad (14)
 \end{aligned}$$

$$\begin{aligned}
 & \text{NLP: } \min x_1^2 + x_2^2 + x_3^2 - 1. \\
 & \text{s.t. } \begin{cases} -x_1^2 + x_2 - x_3^2 \leq 0; \\ x_1 + x_2^2 + x_3 \leq 20; \\ -x_1 - x_2^2 + 2 = 0; \\ x_2 + 2x_3^2 - 3 = 0; \\ x_1, x_2, x_3 \geq 0. \end{cases} \quad (15)
 \end{aligned}$$

针对以上两类规划问题, 分别基于 MATLAB 命令以及智能优化算法进行求解, 结果见表 4. 其中, 用 MATLAB 命令求解时, 对 LP 采用传统单纯形法 (simplex algorithm), 对 NLP 分别采用内点法 (interior-point)、有效集法 (active-set) 及信赖域反射法 (trust-region-reflective) 三种求解算法.

表 4 两类数学规划问题的最优求解结果

求解方式	算法	LP	NLP
MATLAB 命令	simplex algorithm	2.10e+01	-
	interior-point	-	1.42e+00
	active-set	-	2.30e+00
	trust-region-reflective	-	2.05e+00
智能优化方法	MBO	2.10e+01	3.56e-01
	NCSMBO	2.10e+01	5.07e-03

由以上结果可知: 对于 LP 问题, 由于模型相对简单, 两种智能优化方法均可取得与 MATLAB 命令直接求解相一致的结果. 对于 NLP 问题, MBO 与 NCSMBO 获得的最优值明显好于直接求解的结果, 进一步验证了智能优化方法在求解非线性复杂优化问题上的优势. 然而相比之下, 本文 NCSMBO 算法更具有适用性, 最优解要比 MBO 低 2 个数量级, 计算结果更为理想, 这表明在限定进化代数下, NCSMBO 可在有约束解空间内实现有效进化, 探索和开发能力得以充分发挥, 更加擅长对非线性复杂问题的求解.

4 结论

本文通过分析基本帝王蝶算法的进化机制, 提出一种基于非线性云化的自适应帝王蝶算法—NCSMBO. 首先, 在基本帝王蝶算法中引入非线性云化策略, 分别对迁移算子和调整算子的父代个体进行云化, 通过丰富取值空间的方式动态增加种群多样性, 增强高精度搜索能力, 同时为保证收敛性, 云化程度随迭代进行而降低. 其次, 为调和进化后期陷入“早熟”与提高收敛精度之间的矛盾, 进一步提出基于非线性变化方式的调整率. 最后, 基于 12 个标准测试函数对包含 NCSMBO 在内的 7 种算法进行了性能对比分析, 同时结合两类数学规划问题求解实验, 结果均验证了所提算法的可行性和有效性. 后续考虑将 NCSMBO 与工程实践问题相结合, 进一步检验其应用效果.

参考文献 (References)

- [1] Arakawa M, Yamashita Y, Funatsu K. Genetic algorithm-based wavelength selection method for spectral calibration[J]. Journal of Chemometrics, 2011, 25(1): 10-19.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE International Conference on Neural Networks, 1995. IEEE, 2002:1942-1948.
- [3] 陈暄, 孟凡光, 吴吉义. 求解大规模优化问题的改进狼群算法 [J]. 系统工程理论与实践, 2021,41(3):790-808.
(Chen X, Meng F G, Wu J Y. Improved wolf pack algorithm for large-scale optimization problems[J].

- Systems Engineering Theory and Practice, 2021,41(3):790-808.)
- [4] Seyedali M, Andrew L. The whale optimization algorithm[J]. Advances in Engineering Software, 2016,95:51-67.
- [5] 王英聪, 刘军辉, 肖人彬. 基于刺激-响应分工机制的人工蜂群算法[J]. 控制与决策, 2022, 37(4): 881-891. (Wang Y C, Liu J H, Xiao R B. Artificial bee colony algorithm based on stimulus-response labor division[J]. Control and Decision, 2022, 37(4): 881-891.)
- [6] Wang G G, Suash D, Cui Z H. Monarch butterfly optimization[J]. Neural Computing and Applications, 2019, 31(7): 1995-2014.
- [7] Madhusudana R N, Krithivasan K, Gao X Z, et al. Multiobjective hybrid monarch butterfly optimization for imbalanced disease classification problem[J]. International Journal of Machine Learning and Cybernetics, 2019,11:333-361.
- [8] Wang G G, Hao G S, Cheng S, et al. A discrete monarch butterfly optimization for Chinese TSP problem[C]. International Conference on Swarm Intelligence. Bali, Indonesia: Springer, 2016:165-173.
- [9] Feng Y, Yan H, Yu X, et al. A novel monarch butterfly optimization with global position updating operator for large-scale 0-1 knapsack problems[J]. Mathematics, 2019,7(11):811-842.
- [10] Feng Y, Wang G G, Suash D, et al. Solving 0-1 knapsack problem by a novel binary monarch butterfly optimization[J]. Neural Computing and Applications, 2017, 28(7):1619-1634.
- [11] Waheed A H M G, Aman J. Hybridizing artificial bee colony with monarch butterfly optimization for numerical optimization problems[J]. Neural Computing and Applications, 2016, 5: 331-349.
- [12] Ivana S, Marko A, Markovic D, et al. Hybridized monarch butterfly algorithm for global optimization problems[J]. International Journal of Computers, 2018, 3: 63-68.
- [13] 冯艳红, 杨娟, 贺毅朝等. 差分进化帝王蝶优化算法求解折扣 0-1 背包问题 [J]. 电子学报, 2018, 46(6): 1343-1350. (Feng Y H, Yang J, He Y C, et al. Monarch butterfly optimization algorithm with differential evolution for the discounted 0-1 knapsack problem[J]. Acta Electronica Sinica, 2018, 46(6): 1343-1350.)
- [14] 孙林, 赵婧, 徐久成等. 基于改进帝王蝶优化算法的特征选择方法 [J]. 模式识别与人工智能, 2020, 33(11): 981-994. (Sun L, Zhao J, Xu J C, et al. Feature selection method based on improved monarch butterfly optimization algorithm[J]. Pattern Recognition and Artificial Intelligence, 2020, 33(11): 981-994.)
- [15] Hossam F, Ibrahim A, Seyedali M. Improved monarch butterfly optimization for unconstrained global search and neural network training [J]. Applied Intelligence, 2018,48(2):445-464.
- [16] Sun L, Chen S S, Xu J C, et al. Improved monarch butterfly optimization algorithm based on opposition-based learning and random local perturbation[J].Complexity, 2019,6:156-175.
- [17] Abdelmonem M, Mohamed A T. A hybridization of differential evolution and monarch butterfly optimization for solving systems of nonlinear equations[J]. Journal of Computational Design and Engineering, 2019, 6: 354-367.
- [18] Kaviarasan R, Harikrishna P, Arulmurugan A. Load balancing in cloud environment using enhanced migration and adjustment operator based monarch butterfly optimization[J]. Advances in Engineering Software, 2022, 169:1-11.
- [19] Jyotirmayee N, Dash P K, Ranjeeta B. Optimized droop controller based energy management for stand-alone micro-grid using hybrid monarch butterfly and sine-cosine algorithm[J]. Sustainable Energy Technologies and Assessments, 2021, 46:1-14.
- [20] 李德毅, 杜匡. 不确定性人工智能 [M]. 北京: 国防工业出版社, 2005: 224-229. (Li D Y, Du Y. Artificial intelligence with uncertainty[M]. Beijing: National Defense Industry Press,2005:224-229.)
- [21] 张光卫, 何锐, 刘禹等. 基于云模型的进化算法 [J]. 计算机学报, 2008, 31(7) :1082-1091. (Zhang G W, He R, Liu Y, et al. An evolutionary algorithm based on cloud model[J]. Chinese Journal of Computers, 2008, 31(7) :1082-1091.)
- [22] Li D, Liu C, Gan W. A new cognitive model: Cloud model[J]. International Journal of Intelligent Systems, 2009, 24(3):357375.
- [23] Momin J, Xin S Y. A literature survey of benchmark functions for global optimization problems[J]. International Journal of Mathematical Modelling and Numerical Optimization, 2013, 4(2):150-194.
- [24] Sun Y T, Zhang J F, LI G C, et al. Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes [J]. International Journal for Numerical and Analytical Methods in Geomechanics, 2019,4:801-813.

作者简介

李晓平(1989—),女,工程师,博士,从事社会计算、统计学习与数据挖掘等研究, E-mail: lxp11742@163.com;

杜波(1980—),男,副教授,硕士生导师,从事社会计算、危机传播与风险管理的研究, E-mail: tbca_0929@126.com;

王贤文(1982—),男,教授,博士生导师,从事大数据与计算社会学、科学计算学的研究, E-mail: xwang@dlut.edu.cn.