

控制与决策

Control and Decision

具有振荡约束的自然选择萤火虫优化算法

刘景森, 毛艺楠, 李煜

引用本文:

刘景森, 毛艺楠, 李煜. 具有振荡约束的自然选择萤火虫优化算法[J]. *控制与决策*, 2020, 35(10): 2363–2371.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0295>

您可能感兴趣的其他文章

Articles you may be interested in

基于文化萤火虫算法的足球机器人动态路径规划

Cultural firefly algorithm for dynamic path planning of soccer robot

控制与决策. 2018, 33(11): 2015–2020 <https://doi.org/10.13195/j.kzyjc.2017.0821>

自适应分组差分萤火虫算法求解连续空间优化问题

Adaptive grouping difference firefly algorithm for continuous space optimization problems

控制与决策. 2017, 32(7): 1217–1222 <https://doi.org/10.13195/j.kzyjc.2016.0580>

基于萤火虫优化粒子滤波的新型机动目标跟踪算法

Novel target tracking method based on firefly algorithm optimized particle filter

控制与决策. 2017, 32(10): 1758–1766 <https://doi.org/10.13195/j.kzyjc.2016.1157>

融合均匀变异与高斯变异的蝙蝠优化算法

Bat optimal algorithm combined uniform mutation with Gaussian mutation

控制与决策. 2017, 32(10): 1775–1781 <https://doi.org/10.13195/j.kzyjc.2016.1028>

基于GMOGSO的多目标流水车间调度问题

Multi-objective flow shop scheduling problem based on GMOGSO

控制与决策. 2016, 31(10): 1772–1778 <https://doi.org/10.13195/j.kzyjc.2015.1309>

具有振荡约束的自然选择萤火虫优化算法

刘景森^{1,2}, 毛艺楠², 李煜^{3†}

(1. 河南大学 智能网络系统研究所, 河南 开封 475004; 2. 河南大学 软件学院, 河南 开封 475004;
3. 河南大学 管理科学与工程研究所, 河南 开封 475004)

摘 要: 针对基本萤火虫算法高维求解精度低、收敛速度慢、易早熟等缺点, 提出一种具有振荡、约束和自然选择机制的萤火虫算法, 引入二阶振荡因子, 平衡上一代个体对当前代个体的影响, 防止萤火虫个体陷入局部极值; 加入基于 sigmoid 函数的约束因子, 动态调整个体移动距离, 在算法后期避免萤火虫个体在理论最优值附近因过度扰震而导致精度降低的情况; 采用基于高斯积分倒数递减趋势的自然选择, 在保持个体多样性的同时加快算法的收敛速度. 理论分析证明了改进算法的收敛性和时间复杂度. 通过对 10 个不同特征标准测试函数多个维度的函数优化仿真实验, 测试结果表明改进算法的寻优精度和收敛速度均有明显提升, 尤其是在高维情况下, 几乎对于所有函数仍能找到理论最优解, 较好地解决了萤火虫算法不适于高维求解的问题.

关键词: 萤火虫算法; 二阶振荡; sigmoid 函数; 自然选择; 寻优精度; 收敛性

中图分类号: TP301.6 文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0295

引用格式: 刘景森, 毛艺楠, 李煜. 具有振荡约束的自然选择萤火虫优化算法[J]. 控制与决策, 2020, 35(10): 2363-2371.

Natural selection firefly optimization algorithm with oscillation and constraint

LIU Jing-sen^{1,2}, MAO Yi-nan², LI Yu^{3†}

(1. Institute of Intelligent Networks System, Henan University, Kaifeng 475004, China; 2. College of Software, Henan University, Kaifeng 475004, China; 3. Institute of Management Science and Engineering, Henan University, Kaifeng 475004, China)

Abstract: Aiming at the shortcomings of low precision in high dimension, slow convergence speed and precocity in the basic firefly algorithm, the firefly algorithm with oscillation, constraint and natural selection mechanism is proposed. Firstly, the second-order oscillation factor is introduced to balance the influence of the previous generation of individuals on the current generation of individuals, so as to prevent the individuals of fireflies from falling into local extremum. Then, the constraint factor based on sigmoid function is added to dynamically adjust the moving distance of individual, so as to avoid the situation that the individual of firefly caused by excessive disturbance near the theoretical optimal value leads to the reduction of precision in the later stage of the algorithm. Finally, natural selection based on the decreasing trend of reciprocal Gaussian integral is used to keep individual diversity and accelerate the convergence speed of the algorithm. The convergence and time complexity of the improved algorithm are proved by theoretical analysis. Through the simulation of 10 standard functions with multiple dimensions and different characteristics, the test results show that the optimization accuracy and convergence speed of the OCSFA(natural selection firefly optimization algorithm with oscillation and constraint) are obviously improved. Especially in the case of high dimension, the theoretical optimum can still be found for almost all functions, which better solves the problem of the unsuitability of the firefly algorithm for high dimension solution.

Keywords: firefly algorithm; second-order oscillation; sigmoid function; natural selection; optimization accuracy; convergence

0 引 言

随着工程技术领域和科学计算规模的日益增长, 传统优化计算方法无法在合理时间内找到所需

解, 从求解效率来看, 对于这类问题的求解基本无效. 近年来, 基于仿生学的启发式智能优化算法不断得到发展和研究, 因其操作简单、求解高效等特点而

收稿日期: 2019-03-14; 修回日期: 2019-05-04.

基金项目: 河南省重点研发与推广专项项目(182102310886); 河南大学研究生“英才计划”项目(SYL18060145).

责任编辑: 陈家伟.

†通讯作者. E-mail: leey@henu.edu.cn.

受到众多学者的青睐. 例如: 受蚂蚁觅食时利用信息素快速找到目标的行为启发, Colormi 等^[1]提出了蚁群算法 (ant colony algorithm, ACA); 受飞鸟集群活动规律性启发, Kennedy 等^[2]提出了粒子群算法 (particle swarm optimization, PSO); 受自然界花朵授粉过程的启发, Yang^[3]提出了花朵授粉算法 (flower pollination algorithm, FPA); 受蝙蝠利用声呐来探测猎物、绕过障碍物行为的启发, Yang^[4]又提出了蝙蝠算法 (bat algorithm, BA); 受狼群领导层次和狩猎行为的启发, Mirjalili 等^[5]提出了灰狼优化算法 (grey wolf optimizer, GWO); 受座头鲸捕食行为的启发, Mirjalili 等^[6]又提出了鲸鱼算法 (whale optimization algorithm, WOA) 等. 这些仿生算法的提出为解决复杂问题提供了新思路, 并被广泛应用于多个相关领域.

萤火虫算法 (firefly algorithm, FA) 是 Yang^[7] 研究并提出的一种仿生群智能优化算法. 该算法进化机制明晰, 操作简单, 参数设置较少, 低维寻优能力较强, 近些年已成为进化计算领域的重要算法和研究热点之一, 已被应用于预测股票价格区间^[8]、预测太阳辐射^[9]、机器人路径规划^[10]、屏蔽/分流电磁场问题^[11]、地震预测^[12] 以及情绪识别^[13] 等许多领域.

萤火虫算法与其他启发式算法一样, 其本身同样存在对初始解的依赖、易陷入局部极值、后期收敛速度较慢等问题, 尤其是高维求解能力较弱, 影响了其应用的进一步拓展. 对此, 许多学者针对萤火虫算法的不足做了相应改进. Gandomi 等^[14] 提出了在萤火虫种群初始化时引入混沌映射, 加强了算法全局探测能力; Hassanzadeh 等^[15] 提出了一种集合模糊逻辑的萤火虫算法, 使萤火虫受到模糊集合中的萤火虫所吸引, 增强了萤火虫与位置较优萤火虫之间的信息交流, 提高了算法的全局搜索能力; Yu 等^[16] 使用可变策略对步长进行设置, 以解决固定步长易陷入局部最优值的问题, 提高了算法的寻优精度; Verma 等^[17] 提出了基于对立的学习方法对候选解进行初始化, 同时使萤火虫沿着不同维度进行更新, 提高了萤火虫算法的收敛速度; Zhu 等^[18] 提出了一种将动态自适应惯性权重加入位置更新公式的萤火虫算法, 使惯性权重随着目标函数值动态变化, 增强了算法的全局搜索能力, 且不易陷入局部极值; Zhang 等^[19] 提出一种基于回归-成本的吸引力和自适应跳跃的二元运动算子来更新萤火虫位置, 提高了算法的求解性能; Zhng 等^[20] 提出了自适应差分萤火虫算法, 利用自适应分组策略对种群的各子群进行寻优, 以解决萤火虫算法

在寻优过程中出现早熟、求解精度偏低的现象; Wang 等^[21] 提出了一种新的萤火虫算法变体, 让每只萤火虫被临近区域吸引而不是受到整个种群比它更亮的萤火虫所吸引, 从而提高了求解精度并降低了计算的时间复杂度; Xie 等^[22] 提出了一种混合型多目标萤火虫算法, 使用连续决策空间量化的方法生成初始种群, 提高了算法的收敛性、多样性; Kaveh 等^[23] 提出了基于逻辑映射的混沌萤火虫算法, 采用混沌映射策略对吸引力和光强吸收系数进行优化, 提高了算法的鲁棒性和收敛性. 这些改进算法在相应领域的寻优能力均有提升, 但萤火虫算法仍有进一步改进的空间.

本文提出一种具有振荡、约束机制的自然选择萤火虫算法. 首先, 引入二阶振荡因子, 动态改变上一代萤火虫对当前代的影响大小, 增加跳出局部最优的概率, 有利于避免萤火虫陷入局部极值; 然后, 在位置移动中加入基于 sigmoid 函数的约束因子, 在算法后期可以防止萤火虫因过度扰震而导致寻优精度降低现象的发生; 最后, 使用一种基于高斯积分累积分布函数倒数机制的非线性递减趋势自然选择, 加快算法的收敛速度, 同时也能较好地保持萤火虫的多样性. 理论分析证明了改进算法的收敛性和时间复杂度; 而对于多个不同特征、不同维度经典函数进行的测试表明, 改进后算法的收敛速度与寻优精度均有显著提升, 可有效避免早熟现象的发生, 克服萤火虫算法高维求解性能不佳的问题.

1 基本萤火虫算法

萤火虫算法流程如下.

step 1: 初始化算法基本参数, 在求解空间内随机生成各萤火虫的初始位置 $x_i (i = 1, 2, \dots, n)$.

step 2: 根据目标函数计算各萤火虫的适应度值 $f(x_i)$, 并由此求出每个个体的荧光亮度.

step 3: 确定萤火虫 i 与萤火虫 j 间的距离

$$r_{ij} = \|x_j - x_i\| = \sqrt{\sum_{k=1}^d (x_{j,k} - x_{i,k})^2}. \quad (1)$$

其中: d 表示维度, $x_{i,k}$ 为个体 i 在 d 维空间中的第 k 个分量, r_{ij} 表示个体 i 与个体 j 之间的欧氏距离.

step 4: 更新萤火虫 j 的吸引度

$$\beta_{ij} = \beta_0 \times \exp(-\gamma r_{ij}^2). \quad (2)$$

其中: β_0 表示 $r_{ij} = 0$ 时的吸引度, γ 为光强吸收系数. 在萤火虫的视线范围内, 萤火虫的吸引度随着 r_{ij} 的增加而单调递减.

step 5: 更新萤火虫 i 的位置.

萤火虫*i*因受到更亮的萤火虫*j*的吸引而进行位置移动,即

$$x_i(t+1) = x_i(t) + \beta_{ij} \times (x_j(t) - x_i(t)) + \alpha \times (\text{rand} - 0.5). \quad (3)$$

其中: $x_i(t)$ 为第*i*个个体在第*t*代的位置; α 为步长因子,一般被定义为常数; rand 是服从均匀分布的随机因子.位置移动公式可分为3项:第1项反映了当前萤火虫受到上一代萤火虫的位置影响,平衡全局与局部寻优能力;第2项反映了萤火虫群体间的信息传递,是一种信息共享;第3项随机步长的设置可以避免萤火虫陷入局部极值.

step 6: 判断是否满足迭代次数.若是,则输出结果;否则,转step 2继续迭代进化.

2 改进算法OCSFA

2.1 基于振荡和约束机制的改进位置移动

2.1.1 引入二阶振荡因子

在基本萤火虫算法中,萤火虫*i*向更亮的萤火虫*j*移动,有较大的几率到达新的搜索空间,但在求解高维复杂函数时,仍易陷入局部最优,很难从局部极值中跳出.在算法后期,萤火虫之间有较强的相对吸引力,使得萤火虫算法的搜索能力减弱,多样性降低.若不考虑位置移动公式中的第2项(即萤火虫之间的吸引),则萤火虫之间就没有交互.一个种群规模为*n*的群体移动相当于*n*个个体独立运动,如果没有信息交流,则得到最优解的概率很小.为了增加萤火虫群体的多样性,以及加大萤火虫跳出局部最优的概率,本文在萤火虫位置移动公式中引入二阶振荡因子.改进后的位置移动公式如下:

$$x_i(t+1) = \xi \times x_i(t) + \beta_{ij} \times (x_j(t) - x_i(t)) + \alpha \times (\text{rand} - 0.5). \quad (4)$$

$$\xi = \begin{cases} \frac{(2\sqrt{\text{rand}} - 1)(1 + \text{rand})}{\text{rand}}, & t \leq \frac{\max G}{2}; \\ \frac{(2\sqrt{\text{rand}} - 1)\text{rand}}{\text{rand}}, & t > \frac{\max G}{2}. \end{cases} \quad (5)$$

其中: ξ 是由式(5)计算得出的随机数,决定了当前迭代萤火虫位置受上一代位置的影响程度; rand 是服从均匀分布的随机因子; $\max G$ 为最大进化代数.在算法前期 $\xi \geq \frac{2\sqrt{\text{rand}} - 1}{\sqrt{\text{rand}}}$,加大了萤火虫振荡的范围,同时加快了算法的收敛速度.在算法后期,萤火虫大部分聚集在较亮萤火虫的周围,使多样性变差, ξ 的引入可以增加个体的多样性,此时 $\xi < \frac{2\sqrt{\text{rand}} - 1}{\sqrt{\text{rand}}}$,从

而避免了 ξ 过大所造成的萤火虫受上一代影响过大使得最优解影响弱化的问题.

2.1.2 基于sigmoid函数的约束因子

为了提高萤火虫算法的收敛速度,增加对移动位置的约束性,本文在萤火虫位置公式中加入基于sigmoid函数的约束因子.结合2.1.1节,新的萤火虫位置更新公式为

$$x_i(t+1) = \varphi(t) \times (\xi \times x_i(t) + \beta_{ij} \times (x_j(t) - x_i(t)) + \alpha \times (\text{rand} - 0.5)), \quad (6)$$

$$\varphi(t) = \frac{1}{\left(1 + \exp\left(\frac{(\ln 3 + \ln 99) \times t - \ln 99}{\max G}\right)\right)}. \quad (7)$$

其中: $\varphi(t)$ 为约束因子, $\varphi(t)$ 随着进化步数的增加非线性递减; t 为当前进化次数; $\max G$ 为最大进化次数.经多次实验发现, $\varphi(t) \in [0.45, 0.99]$ 时算法收敛速度最快. $t = 0$ 时, $\varphi(t)$ 取值为0.99,在算法前期不影响萤火虫*i*向周围较亮萤火虫*j*移动的速度;在后期,经过前期萤火虫种群的移动,大部分萤火虫聚集在最优萤火虫周围,距离最优萤火虫很近,此时 $\varphi(t)$ 取值较小,在减小 ξ 影响的同时也可以避免在理论最优值附近因过度扰震而导致精度降低.

2.2 自然选择机制

将自然选择机制与引入二阶振荡和约束因子的萤火虫算法相结合.自然选择的基本思想是在每次迭代循环中将群体中的萤火虫按照适应度值排列,以群体中适应度最优的 $1/m$ 萤火虫的位置替换适应度最差的 $1/m$ 萤火虫的位置,从而提高萤火虫接近最优萤火虫的比例,加快收敛速度.本文将*m*的取值结合高斯积分累积分布函数,在函数优化实验中取得了很好的效果,*m*取值公式如下:

$$m = m_{\max} \times \text{CDF}(t, \text{mu}, \text{sigma}). \quad (8)$$

其中: m_{\max} 是*m*的最大值,经多次实验发现, $m_{\max} = 22$ 时算法寻优性能最好,将*m*向上取整; $\text{CDF}(t, \text{mu}, \text{sigma})$ 为累积分布函数,这里*t*为当前进化次数, mu 为均值, sigma 为标准差.为了防止*m*值过小导致替换数量过大而损害多样性,经多次实验测试发现,当*m*从数值为5开始递增时,算法的收敛效果最好,因此本文 sigma 取值为600.因*m*呈非线性递增趋势,故 $1/m$ 是非线性递减的.在算法前期,替换稍多一些位置差的萤火虫,有利于加快萤火虫向最优值移动,增加萤火虫之间正确信息的传递;在算法后期,萤火虫大部分聚集在一起,替换较少位置差的萤火虫,有利于保持萤火虫群体的多样性.

2.3 OCSFA算法流程

OCSFA算法描述如下:

适应度函数 $f(x)$, $x = (x_1, \dots, x_d)^T$
 萤火虫种群的初始化 $x_i = (1, 2, \dots, n)$
 各参数的初始化 γ 、 $\max G$ 、 α 、 β_0
 由 $f(x)$ 计算萤火虫 x_i 的荧光亮度 I_i
 while($t < \max G$)
 由式(5)计算二阶振荡因子
 按照式(7)计算约束因子
 根据式(8)计算 m 的值, 并进行自然选择
 for $i = 1 : n$
 for $j = 1 : i$
 计算相对吸引度 β_{ij}
 计算萤火虫间的距离 r_{ij}
 if ($I_j > I_i$)
 在 d 维空间, 通过式(6)使萤火虫 i 向萤火虫 j 进行移动
 end if
 end for j
 end for i
 按适应度值对所有萤火虫排序, 确定当前最优位置
end while
输出结果

3 OCSFA算法的理论分析

3.1 收敛性分析

收敛性是评估算法稳定性和有效性的一个关键因素, 随着进化步数的增加, 寻优结果与理论最优值之间的误差越来越小, 最终趋近于一个固定值附近. 在基本萤火虫算法中, 搜索部分体现于位置更新公式(3), 决定了算法的收敛能力. 同样, 在分析本文改进算法 OCSFA 的收敛性时, 着眼点也应放在位置更新公式(6)上. 在群智能算法研究中, 学者们采用了一些不同的方法对一些算法的收敛性进行分析. 文献[24-25]利用二阶常系数非齐次差分方程分别对万有引力算法和蝙蝠算法的收敛性进行了相应分析, 本文也采用同样的方法对 OCSFA 进行收敛性分析.

由改进算法 OCSFA 的位置更新公式(6)可以看出, $x(t)$ 是多维变量, 但每一维之间是相互独立互不影响的, 因此, 可以将其简化到对一维进行分析. 为了计算方便, 可以假设对于当前迭代种群, 取值或计算量保持不变的量为常量. 由此, 设位置更新公式(6)中的 ξ 为常量 a , $\varphi(t)$ 为常量 b , 吸引度 β_{ij} 为常量 c , 第3项干扰因子 $\alpha \times (\text{rand} - 0.5)$ 为常量 d ; 第 t 代较亮个体 x_j 的位置记作 g_a , 第 $t+1$ 代较亮个体 x_j 的位置记作 p_b .

由此, 式(6)可转化为

$$\begin{aligned} x(t+1) &= b(ax(t) + c(g_a - x(t)) + d), \quad (9) \\ x(t+2) &= \end{aligned}$$

$$b(ax(t+1) + c(p_b - x(t+1)) + d). \quad (10)$$

由式(9)和(10)可得

$$\begin{aligned} x(t+2) + (1+r)x(t+1) + rx(t) &= \\ 2bd + bcg_a + bcp_b, \end{aligned} \quad (11)$$

其中 $r = bc - ab$.

根据式(11)的二阶常系数非齐次差分方程, 可求得其特征方程为

$$\lambda^2 + (1+r)\lambda + r = 0.$$

于是 $\Delta = (1+r)^2 - 4r = (1-r)^2 \geq 0$, 因此, 本文算法收敛过程需要满足以下两种情况:

1) 当 $\Delta = 0$ 时, 特征方程具有两个相同的实根, 计算可得 $\lambda_1 = \lambda_2 = -\frac{1+r}{2} = -1$, 此时有 $x(t) = (A_0 + A_1 t)\lambda^t$, A_0, A_1 为待定系数, 由 $x(0)$ 来确定. 经计算得到

$$\begin{cases} A_0 = x(0), \\ A_1 = (r-1)x(0) - bcg_a - bc. \end{cases}$$

2) 当 $\Delta > 0$ 时, 特征方程具有两个不同的实根, 计算可得 $\lambda_{1,2} = \frac{-(1+r) \pm \sqrt{\Delta}}{2}$, 此时 $x(t) = A_0 + A_1 \lambda_1^t + A_2 \lambda_2^t$, A_0, A_1 和 A_2 为待定系数, 由 $x(0)$ 、 $x(1)$ 和 $x(2)$ 确定. 经计算得到

$$\begin{cases} A_0 = x(0) - A_1 - A_2, \\ A_1 = \frac{\lambda_2 x(0) - (1 + \lambda_2)x(1) + x(2)}{(\lambda_2 - \lambda_1)(1 - \lambda_1)}, \\ A_2 = \frac{\lambda_1 x(0) - (1 + \lambda_1)x(1) + x(2)}{(\lambda_1 - \lambda_2)(1 - \lambda_2)}. \end{cases}$$

其中

$$\begin{aligned} x(1) &= b(ax(0) + c(g_a - x(0)) + d), \\ x(2) &= b(cg_a + d - rx(0)). \end{aligned}$$

基于上述分析, 要使算法迭代收敛, 必须满足下面两个条件: 1) 当 $t \rightarrow \infty$ 时, $x(t)$ 能够逼近于某一特定值; 2) $\|\lambda_1\| < 1$ 且 $\|\lambda_2\| < 1$ ^[25].

计算可得如下结论:

1) 当 $\Delta = 0$ 时, 收敛区域为 $r = 1$.

2) 当 $\Delta > 0$ 时, $\left| \frac{-(1+r) \pm (1-r)}{2} \right| < 1$, 即 $-1 < r \leq 1$, 其中 $r = bc - ab$. 因此, 可以得出 $-1 < bc - ab \leq 1 \Rightarrow -1 < b(c-a) \leq 1 \Rightarrow -\frac{1}{b} < c-a \leq \frac{1}{b}$, 则有

$$\begin{cases} c-a \leq \frac{1}{b}, \\ c-a > -\frac{1}{b}; \end{cases} \Rightarrow \begin{cases} a \geq c - \frac{1}{b}, \\ a < c + \frac{1}{b}. \end{cases}$$

由2.1.2节可知约束因子 $\varphi(t) \in [0.45, 0.99]$, 故 b 也在此区间, 因此, $1/b$ 可在区间 $(1, 2.2]$ 之间任意取值. 又因为 c 的取值范围为 $(0, 1)$, 所以 $c - 1/b$ 的取值

范围为 $(-2.2, 0)$, $c + 1/b$ 的取值范围为 $(1, 3.2)$. 经计算可得 a 、 b 的收敛域分别为 $-2.2 \leq a < 3.2$, $-1 < b \leq 1$. 然而, 在本文算法中 $a \in (-1, 2)$, $b \in [0.45, 0.99]$, 因此, 振荡因子和约束因子的取值范围均符合收敛区域, 所以改进后的算法收敛.

3.2 时间复杂度分析

时间复杂度是体现算法性能的关键因素, 它反映了算法的运算效率. 文献[26-27]分别对布谷鸟算法的时间复杂度进行了分析, 本文采用同样思想对OCSFA算法的时间复杂度进行分析.

在基本萤火虫算法中, 假设种群大小为 N , 个体维度为 n , 各参数(最大吸引度、光吸引系数、步长因子)设置的时间为 η_0 , 产生均匀分布随机数的时间为 η_1 , 求给定适应度函数的时间为 $f(n)$, 则种群初始化阶段的时间复杂度为

$$O(\eta_0 + N(n\eta_1 + f(n))) = O(n + f(n)).$$

设按萤火虫适应度值排序所需时间为 η_2 , 计算萤火虫 i 与更亮萤火虫 j 间距离的时间为 η_3 , 求解萤火虫 j 对萤火虫 i 吸引度的执行时间为 η_4 , 萤火虫个体每一维产生随机步长所需时间为 η_5 , 执行个体每一维位置移动需要时间为 η_6 , 则该阶段的时间复杂度为

$$O(N(\eta_2 + \eta_3 + \eta_4 + n\eta_5 + n\eta_6)) = O(n).$$

由上述可得, 基本算法求解每代最优解的总时间复杂度为

$$T(n) = O(n + f(n)) + O(n) = O(n + f(n)).$$

在OCSFA算法中, 算法的种群大小、参数设置时间、维度、求解适应度值时间均与基本萤火虫算法相同. 因此, OCSFA算法在种群初始化阶段的时间复杂度与基本萤火虫算法相同, 其时间复杂度为

$$O(\eta_0 + N(n\eta_1 + f(n))) = O(n + f(n)).$$

分析改进萤火虫算法OCSFA的流程. 设计算二阶振荡因子和约束因子的时间分别为 t_1 和 t_2 , 计算 m 和种群替换的时间分别为 t_3 和 t_4 , 执行萤火虫每一维位置移动的所需时间为 t_5 , 则改进后算法在该阶段的时间复杂度为

$$O(N(\eta_2 + t_1 + t_2 + t_3 + t_4 + \eta_3 + \eta_4 + n\eta_5 + nt_5)) = O(n).$$

基于上述分析, OCSFA算法寻找每代最优值的总时间复杂度为

$$O(n + f(n)) + O(n) = O(n + f(n)).$$

由以上分析可以看出, 本文改进算法OCSFA与基本萤火虫算法相比, 时间复杂度并未发生改变, 算法的运行效率也没有降低.

4 仿真实验

为了全面检验本文改进算法的寻优能力, 选取10个具有不同寻优特征的经典测试函数, 将本文算法与基本萤火虫算法(FA)、蝙蝠算法(BA)、a variable step size firefly algorithm for numerical optimization (VSSFA)^[16]、动态自适应权重的萤火虫算法(IWFA)^[18]和chaotic firefly algorithms based on Logistic map (CLFA)^[23]共6种算法, 分别在10维、50维和100维上进行对比测试.

4.1 测试函数

10个经典测试函数如表1所示, $f_1(x)$ 、 $f_2(x)$ 、 $f_5(x)$ 、 $f_8(x)$ 和 $f_9(x)$ 为多维单峰函数, $f_3(x)$ 、 $f_4(x)$ 、 $f_6(x)$ 、 $f_7(x)$ 和 $f_{10}(x)$ 为多维多峰函数. 其中: $f_3(x)$ 是标准多维多峰函数, 自变量互相影响; $f_4(x)$ 、 $f_7(x)$ 和 $f_{10}(x)$ 函数有大量局部最优值; $f_6(x)$ 函数有大量局部最优值和障碍物. 这些函数求解难度较高, 很适于测试算法的求解能力和寻优性能.

4.2 寻优精度分析

为了测试改进算法OCSFA的寻优性能, 本文将测试函数 $f_1(x) \sim f_{10}(x)$ 的维度分别设置为 $d = 10/50/100$. 为保证实验的公平性与客观性, 6种算法均在相同条件下运行50次, 4种萤火虫改进算法的参数设置也都与基本萤火虫算法(FA)保持一致. 各算法参数设置如下.

FA参数设置为: 光强吸收系数为 $\gamma = 1.0$, 最大吸引度为 $\beta_0 = 1.0$, 步长因子 $\alpha = 0.5$; BA参数设置为: 频率的最大和最小值分别为 $Q_{\min} = 0$, $Q_{\max} = 2$, 响度最大值 $A_0 = 0.25$, 脉冲最大值 $r_0 = 0.5$; OCSFA算法参数设置为: 自然选择倒数 m 的最大值 $m_{\max} = 22$, 光强吸收系数为 $\gamma = 1.0$, 最大吸引度为 $\beta_0 = 1.0$, 步长因子 $\alpha = 0.5$; VSSFA参数设置为: 光强吸收系数为 $\gamma = 1.0$, 最大吸引度为 $\beta_0 = 1.0$, 步长因子 $\alpha = 0.5$; IWFA参数设置为: 光强吸收系数为 $\gamma = 1.0$, 最大吸引度为 $\beta_0 = 1.0$, 步长因子 $\alpha = 0.5$; CLFA参数设置为: 最大吸引度为 $\beta_0 = 1.0$, 步长因子 $\alpha = 0.5$, 逻辑映射中常数 $r = 4.0$.

6种算法的种群大小均为100, 最大进化代数 $\max G = 1000$. 表2统计了不同维数下运行50次得到的最优解、最差解和平均值.

由表2中 $d = 10$ 的寻优结果可以看出, 在10维条件下, 除函数 $f_4(x)$ 、 $f_7(x)$ 外, FA、CLFA、VSSFA和OCSFA都表现出较好的寻优精度, 找到的平均值均优于BA和IWFA算法, 且在函数 $f_1(x) \sim f_3(x)$ 、 $f_5(x)$ 、 $f_8(x)$ 、 $f_9(x)$ 、 $f_{10}(x)$ 中, 找到的最优值、最差值和平均值都为全局最优解, 表现出萤火虫自身机制在低维条件下的优越性能. OCSFA算法在这10个

表1 测试函数

函数名称	函数	最优值
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	0
Axis parallel hyper-ellipsoid	$f_2(x) = \sum_{i=2}^d ix_i^2$	0
Griewank	$f_3(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0
Rastrigrin	$f_4(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$	0
Rotated hyper-ellipsoid	$f_5(x) = \sum_{i=1}^d \sum_{j=1}^i x_j^2$	0
Ackley	$f_6(x) = -20 \times \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	0
Inverted cosine wave	$f_7(x) = -\sum_{i=1}^{d-1} \left(\exp\left(-\frac{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}{8}\right) \times \cos(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}})\right)$	$-d + 1$
Sum squares	$f_8(x) = \sum_{i=1}^d ix_i^2$	0
Schwefel's problem 2.22	$f_9(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	0
Alpine	$f_{10}(x) = \sum_{i=1}^d x_i \sin(x_i + 0.1x_i) $	0

函数中的整体表现更加优异,对于函数 $f_6(x)$, OCSFA 的求解精度与 FA、CLFA 和 VSSFA 相当,远高于 BA 和 IWFA;而对于另外9个函数, OCSFA 不管是找到的最优解、平均值还是最差解都是理论最优值. BA 和 IWFA 算法在给定的10个函数中,平均值均没有达到全局最优值;FA、CLFA 和 VSSFA 算法在 $f_4(x)$ 和 $f_7(x)$ 两个函数中平均值的精度远远低于 OCSFA 算法. 因此, OCSFA 算法找到理论最优值的比例明显大于 FA、BA、CLFA、VSSFA、IWFA,且寻优精度和稳定性更好.

表2中 $d = 50$ 和 $d = 100$ 的寻优结果显示,在50维和100维条件下, OCSFA 的寻优精度明显优于 FA、BA、CLFA、VSSFA 和 IWFA, OCSFA 在除函数 $f_6(x)$ 之外其他9个函数中所找到的最优解、最差解和平均值都是理论最优值,而其他5种算法则在这10个函数中都无法求解到理论最优值,表现出本文改进算法 OCSFA 优越的高维寻优性能.

对于函数 $f_1(x) \sim f_{10}(x)$,其他5种算法的寻优精度都会随着维度的增加有所降低,而 OCSFA 算法则不受维度变化的影响. 在50维和100维条件下,BA、FA、CLFA、VSSFA 和 IWFA 算法也无法找到全局最优解,尤其在函数 $f_1(x)$ 、 $f_2(x)$ 、 $f_4(x)$ 、 $f_5(x)$ 和 $f_8(x) \sim f_{10}(x)$ 中,FA 和 VSSFA 算法的最优值远远大于理论最优值,暴露出萤火虫算法原有机制高维求解精度低这一缺陷. 而 OCSFA 算法却仍然能够在 $f_1(x) \sim f_5(x)$ 和 $f_7(x) \sim f_{10}(x)$ 这9个函数中找到理

论最优值;在函数 $f_6(x)$ 中, OCSFA 虽然没有找到全局最优值0,但找到的最优解、最差解和平均值均优于其他5种算法. 综上所述,本文改进算法 OCSFA 提高了在高维条件下的寻优精度.

仿真结果表明,在10维、50维和100维的条件下, OCSFA 算法表现出较好的求解性能,主要是因为萤火虫位置移动公式中引入二阶振荡因子以及基于 sigmoid 函数的约束因子,有利于萤火虫算法在高维情况下从局部极值中跳出,同时也防止了在算法后期由于移动距离过大而产生扰震现象,有利于萤火虫逐步移动到最亮萤火虫的位置,提高了算法的寻优精度,增加了找到理论最优值的几率. OCSFA 在高维时几乎对所有函数仍能找到理论最优解,这充分表明了这些改进较好地解决了萤火虫算法不适合于高维求解的问题.

4.3 收敛曲线分析

一个算法性能的优劣可以直观地通过收敛曲线展现出来,收敛曲线显示了算法陷入局部最优中的次数和收敛速度. 由于多维多峰函数更为复杂,易使算法陷入局部极值,这些函数的收敛情况更能说明算法本身的寻优能力. 因此,下面只给出10个测试函数中所有5个多维多峰函数的收敛曲线,而其余5个多维单峰函数的收敛曲线对比也表现出同样的结果,不再冗赘列出. 图1~图5是 FA、BA、CLFA、IWFA、VSSFA 和 OCSFA 在多维多峰函数 $f_3(x)$ 、 $f_4(x)$ 、 $f_6(x)$ 、 $f_7(x)$ 和 $f_{10}(x)$ 上维度 $d = 100$ 时的收敛情况.

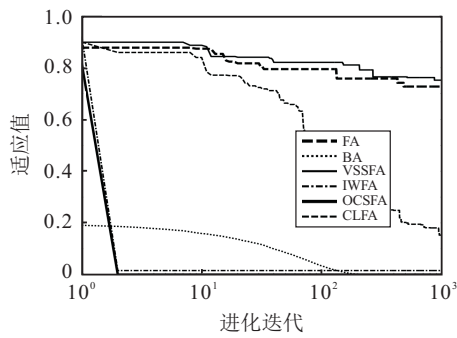


图1 $f_3(x)$ 函数的收敛曲线

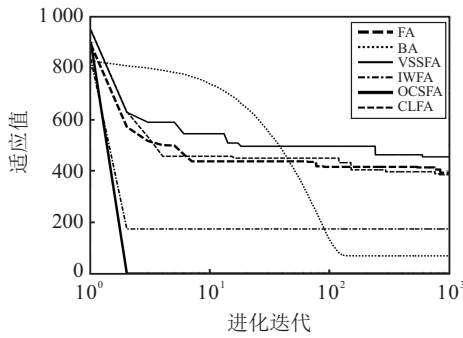


图2 $f_4(x)$ 函数的收敛曲线

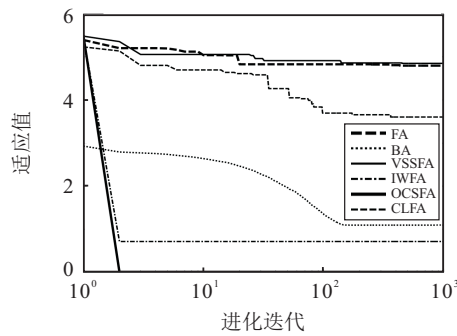


图3 $f_6(x)$ 函数的收敛曲线

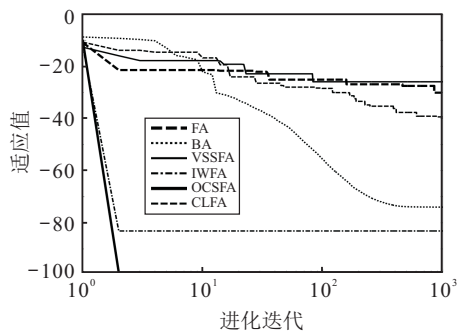


图4 $f_7(x)$ 函数的收敛曲线

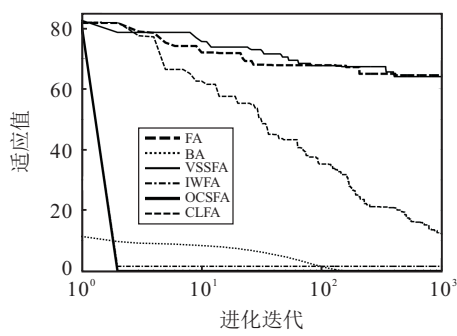


图5 $f_{10}(x)$ 函数的收敛曲线

上面5个多维多峰函数的收敛曲线清晰地展现了FA、BA、CLFA、VSSFA、IWFA和OCSFA算法在进化过程中适应度值的变化趋势.由图1~图5可以看出,OCSFA在所有5个多维多峰函数上都能以更快的速度收敛到全局最优值.

在图1和图5中,本文改进算法OCSFA与IWFA收敛速度同样快,但IWFA在后期陷入到局部极值中无法跳出,从4.2节的表2中可以看出:IWFA在给定1000代进化步数中无法找到理论最优值;BA算法在迭代1000次内可以收敛到最优值附近,较为贴合理论最优,但收敛速度较慢,若减少进化次数,则BA无法收敛到理论最优值附近;FA和VSSFA收敛性较差,无法在1000代内收敛到最优值附近;CLFA在算法后期仍有下降趋势,但速度较慢,无法在给定的进化次数内收敛到全局最优值.在图2~图4中,IWFA算法前期就已经陷入到局部极值中,无法跳出;VSSFA和FA算法收敛速度十分缓慢,在给定进化代数内完全无法收敛到全局最优解;虽然BA的收敛曲线在后期仍有下降趋势,但在1000代内无法收敛到全局最优解;只有OCSFA能够快速收敛到理论最优值.

由上述分析可知,OCSFA在高维条件下寻优精度和收敛速度明显优于其他5种算法,这主要是因为萤火虫位置更新中引入二阶振荡因子,使萤火虫在陷入局部最优时有极大的跳出概率,并增加了萤火虫的搜索范围;而基于sigmoid函数压缩因子的加入,可以避免萤火虫在最优值附近产生扰震现象,从而提高了求解精度,增加了找到理论最优解的可能;结合高斯积分自然选择机制的引入,则加快了萤火虫的寻优速度,增加了种群中正确信息所占比例,使其能在较少进化代数中就可以找到理论最优值.

综上所述,本文提出的OCSFA算法不管是在低维条件下还是在高维条件下都具有较好的进化寻优性能,其求解精度、收敛速度和稳定性均优于FA、BA、CLFA、VSSFA和IWFA五种对比算法.

5 结论

本文针对基本萤火虫算法高维求解精度不高、易陷入局部最优以及收敛速度较慢等不足进行了改进.首先,引入了二阶振荡因子,平衡上一代个体的影响,提高萤火虫跳出局部最优值的几率;其次,给出一种基于sigmoid函数自适应递减的压缩因子,避免萤火虫在极值点附近扰震现象的产生;再次,加入基于高斯积分累积分布函数的自然选择机制,使淘汰替换数量随进化次数非线性下降,在保证种群多样性的同时加快了萤火虫向最亮个体的移动,提高了算法的收敛速度;最后,通过理论分析证明了改进算法的收敛性和时间复杂度.通过对6种算法在10个不同特

征测试函数上计算结果及其中5个多峰函数收敛曲线的对比分析,验证了本文改进算法的有效性和优越性。

参考文献(References)

- [1] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[C]. Proceedings of European Conference on Artificial Life. Paris: Elsevier Press, 1991: 134-142.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proceedings of IEEE International Conference on Neural Networks. Perth: IEEE, 1995: 1942-1948.
- [3] Yang X S. Flower pollination algorithm for global optimization[M]. Berlin: Springer, 2012: 242-243.
- [4] Yang X S. Bat algorithm for multi-objective optimisation[J]. International Journal of Bio-Inspired Computation, 2011, 3(5): 267-274.
- [5] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3): 46-61.
- [6] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95(5): 51-67.
- [7] Yang X S. Firefly algorithms for multimodal optimization[C]. International Symposium on Stochastic Algorithms. Berlin, Heidelberg: Springer, 2009: 169-178.
- [8] Xiong T, Bao Y K, Hu Z Y. Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting[J]. Knowledge-Based Systems, 2014, 55: 87-100.
- [9] Olatomiwa L, Mekhilef S, Shamshirband S, et al. A support vector machine-firefly algorithm-based model for global solar radiation prediction[J]. Solar Energy, 2015, 115: 632-644.
- [10] Xue H, Shao Z P, Pan J C, et al. Cultural firefly algorithm for dynamic path planning of soccer robot[J]. Control and Decision, 2018, 33(11): 2015-2020.
- [11] Alb M, Alotto P, Magele C, et al. Firefly algorithm for finding optimal shapes of electromagnetic devices[J]. IEEE Transactions on Magnetics, 2016, 52(3): 1-4.
- [12] Akhoondzadeh M. Firefly algorithm in detection of TEC seismo-ionospheric anomalies[J]. Advances in Space Research, 2015, 56(1): 10-18.
- [13] Zhang L, Mistry K, Neoh S C, et al. Intelligent facial emotion recognition using moth-firefly optimization[J]. Knowledge-Based Systems, 2016, 111: 248-267.
- [14] Gandomi A H, Yang X S, Talatahari S, et al. Firefly algorithm with chaos[J]. Communications in Nonlinear Science & Numerical Simulation, 2013, 18(1): 89-98.
- [15] Hassanzadeh T, Kanan H R. Fuzzy FA: A modified firefly algorithm[J]. Applied Artificial Intelligence, 2014, 28(1): 47-65.
- [16] Yu S H, Zhu S L, Ma Y, et al. A variable step size firefly algorithm for numerical optimization[J]. Applied Mathematics and Computation, 2015, 263(C): 214-220.
- [17] Verma O P, Aggarwal D, Patodi T. Opposition and dimensional based modified firefly algorithm[J]. Expert Systems with Applications, 2016, 44(C): 168-176.
- [18] Zhu Q G, Xiao Y K, Chen W D, et al. Research on the improved mobile robot localization approach based on firefly algorithm[J]. Chinese Journal of Scientific Instrument, 2016, 37(2): 323-329.
- [19] Zhang Y, Song X F, Gong D W. A return-cost-based binary firefly algorithm for feature selection[J]. Information Sciences, 2017, 418/419: 561-574.
- [20] Zhang Q, Li P C. Adaptive grouping difference firefly algorithm for continuous space optimization problem[J]. Control and Decision, 2017, 32(7): 1217-1222.
- [21] Wang H, Wang W, Zhou X, et al. Firefly algorithm with neighborhood attraction[J]. Information Sciences, 2017, 382/383: 374-387.
- [22] Xie C W, Xiao C, Ding L X, et al. HMOFA: A hybrid multi-objective firefly algorithm[J]. Journal of Software, 2018, 29(4): 1143-1162.
- [23] Kaveh A, Javadi S M. Chaos-based firefly algorithms for optimization of cyclically large-size braced steel domes with multiple frequency constraints[J]. Computers & Structures, 2019, 214: 28-39.
- [24] Liu J S, Xing Y H, Li Y. A gravitational search algorithm with adaptive mixed mutation for function optimization[J]. International Journal of Performability Engineering, 2018, 14(4): 681-690.
- [25] Li Z Y, Ma L, Zhang H Z. Convergence analysis of bat algorithm[J]. Mathematics in Practice and Theory, 2013, 43(12): 182-190.
- [26] Zhang Y W, Wang L, Wu Q D. Dynamic adaptation cuckoo search algorithm[J]. Control and Decision, 2014, 29(4): 617-622.
- [27] Zhou H, Li Y. Cuckoo search algorithm with dynamic inertia weight[J]. CAAI Transactions on Intelligent Systems, 2015, 10(4): 645-651.

作者简介

刘景森(1968—),男,教授,博士,从事智能算法、优化控制和网络安全等研究, E-mail: ljs@henu.edu.cn;

毛艺楠(1993—),女,硕士生,从事智能算法的研究, E-mail: mynhenu@163.com;

李煜(1969—),女,教授,博士,从事智能算法和电子商务等研究, E-mail: leey@henu.edu.cn.

(责任编辑:李君玲)