文章编号: 1001-0920(2017)07-1217-06 DOI: 10.13195/j.kzyjc.2016.0580

自适应分组差分萤火虫算法求解连续空间优化问题

张 强†, 李盼池

(东北石油大学 计算机与信息技术学院，黑龙江 大庆 163318)

摘 要: 提出一种自适应分组差分萤火虫算法求解连续空间优化问题. 利用自适应分组策略对种群进行分子群寻优,基于均匀设计理论调整算法参数,通过云模型算法来改进最优个体的随机扰动行为,引入个体能效吸引力来改进非最优个体更新方式. 最后,利用差分变异算法和混沌理论完成个体变异. 典型复杂函数测试表明,所提出的算法具有很好的收敛精度和计算速度.

关键词: 萤火虫算法；云模型；均匀设计；混沌；连续空间优化

中图分类号: TP301.6 **文献标志码:** A

Adaptive grouping difference firefly algorithm for continuous space optimization problems

ZHANG Qiang†, LI Pan-chi

(School of Computer and Information Technology，Northeast Petroleum University，Daqing 163318, China)

Abstract: An adaptive grouping difference firefly algorithm is proposed to solve the continuous space optimization problem. In the algorithm, the population is divided into subgroups for optimization based on adaptive grouping strategy and the parametors are adjusted based on the uniform design theory the stochastic perturbation behavior of optimal individuals is updated by using the cloud model algorithm. The updating method of the non-optimal individual is improved by introducing the individual energy efficiency attraction. Finally, the individual variation is completed by using the differential mutation algorithm and chaos theory. Simulation results of the typical complex functions show that the algorithm has good the convergence precision and computing speed.

Keywords: firefly algorithm；cloud model；uniform design；chaos；continuous space optimization

0 引 言

萤火虫算法(FA)[1]是一种群体搜索随机优化算法,目前已应用到智能优化、阈值选择和图像处理等[2-6]领域. FA寻优结果对初始解具有一定的依赖性,也存在易陷于局部最优和后期收敛速度慢等问题. 究其原因是由于每个萤火虫在其搜索范围内不断向种群中最亮的个体飞行,通过位置的迭代找寻到全局或局部最优解,若搜索范围内没有更亮个体,萤火虫则作随机运动,这样就降低了发现率和收敛速度. 已有学者在个体进化、种群多样性、启发式算法融合和混沌变异等方面提出了一些改进方法[7-13].

本文提出一种自适应分组差分萤火虫算法(AGD-FA). 首先采用佳点集理论对萤火虫种群进行初始化,利用自适应分组策略对种群进行分子群寻优,通过云模型算法来改进萤火虫最优个体的随机扰动行为,引入个体能效吸引力来改进萤火虫个体更新方式,即在其寻优行为中考虑自身个体能效,以求消耗更低的能效来获取更多的能量;最后利用反向学习理论和混沌理论完成个体变异.

1 基本萤火虫算法原理

萤火虫算法由亮度和吸引度构成. 利用亮度来表示萤火虫个体的优劣,进而控制个体的移动方向,利用吸引度来控制个体的移动距离,通过不断更新这两个要素进行寻优. 算法描述如下:

1) 萤火虫的相对荧光亮度为

$$I = I_0 \times \mathrm{e}^{-\gamma r_{ij}}. \tag{1}$$

其中: I_0为最大萤光亮度,由目标适应度决定,适应度

收稿日期: 2016-05-10；**修回日期:** 2016-07-15.

基金项目: 国家自然科学基金项目(61170132, 61502094)；黑龙江省自然科学基金项目(F2015020)；黑龙江省教育厅项目(12541086).

作者简介: 张强(1982−), 男, 副教授, 博士, 从事智能算法、神经网络等研究；李盼池(1969−), 男, 教授, 博士后, 从事量子智能优化、过程神经网络等研究.

†通讯作者. E-mail: dqpi_zq@163.com

越优亮度越高;γ为光强吸收系数(通常为常数);r_{ij}为萤火虫i与j之间的空间距离.

2) 萤火虫的吸引度为

$$\beta = \beta_0 \times \mathrm{e}^{-\gamma r_{ij}2}. \tag{2}$$

其中:β_0为最大吸引度(通常为常数),γ为光强吸收系数,r_{ij}为萤火虫i与j之间的空间距离.

3) 萤火虫i被萤火虫j吸引后,按下式更新位置:

$$x_i = x_i + \beta \times (x_j - x_i) + \alpha \times \left(\mathrm{rand}() - \frac{1}{2}\right). \tag{3}$$

其中:x_i, x_j为萤火虫i和j所处的空间位置;$\alpha \in [0,1]$为步长因子;rand() 为$[0,1]$上服从均匀分布的随机数.

2 自适应分组差分萤火虫算法原理

2.1 基于佳点集理论的种群初始化

萤火虫算法从随机初始化萤火虫个体开始迭代寻优,若存在某些个体在最优解附近,则在一定程度上能够加速算法收敛和提高寻优性能. 佳点集理论[14]已证明,近似计算函数在s维欧氏空间单位立方体上积分时,用n个佳点构成的加权和比采用任何其他n个点所得到的误差都要小. 已有学者[15-16]将其应用到种群初始化,取得了较好的寻优效果,具体定义如下:

设G_s是s维欧氏空间中的单位立方体,若$r \in G_s$,形为$p_n(k) = \{(\{r_1^{(n)} * k\}, \{r_2^{(n)} * k\}, \cdots, \{r_s^{(n)} * k\}), 1 \leqslant k \leqslant n\}$,其偏差$\varphi(n)$满足$\varphi(n) = C(r,\varepsilon)n^{-1+\varepsilon}$,则称$p_n(k)$为佳点集,$r$为佳点. 其中$C(r,\varepsilon)$为只与$r$和$\varepsilon(\varepsilon > 0)$有关的常数,一般情况下,取$r = \{2\cos(2\pi k/p), 1 \leqslant k \leqslant s\}$,$p$为满足$(p-s)/2 \geqslant s$的最小素数.

2.2 自适应分组策略

自然界中萤火虫的活动范围具有一定领域性,子种群规模在某一时刻也相对固定. 随着不断地飞行或被其他个体吸引,其活动范围、规模和子群个数会随之发生变化. 分析FA在求解高维多峰函数时的运动轨迹发现:存在若干个体某一时刻在全局最优值附近内迭代进化,但由于它们的适应度不如已找到的局部最优解,经过一段时间的进化后,这些个体存在被局部最优解从全局最优范围内吸引走的可能性,造成快速陷入局部最优. 如果进化过程中采用局部邻域,则可有效避免这种可能性. 基于自然界中萤火虫的领域特性,提出一种自适应分组策略来模拟这种特性,进而避免算法快速陷入局部最优. 分组方式如下:

1) 将整个萤火虫种群分成m个子群,每个子群包含n个萤火虫.

2) 将萤火虫种群内的个体按适应值降序排列,前m个萤火虫依次分配给m个子群,则第$m+1$个萤火虫进入第1个子群.

3) 每迭代一次均计算子群内萤火虫个体的方差. 用方差来衡量子群内个体的离散程度,方差越小说明离散程度越小,表明或是找到最优解,或是陷入局部最优.

4) 递减子群个数m. 令$D_m^t(x) = (D_1^t(x), D_2^t(x), \cdots, D_m^t(x))$为第$t$次迭代后的方差,$\alpha_i^t(x) = D_i^t(x) \Big/ \sum_{i=1}^{m} D_i^t(x), 1 \leqslant i \leqslant m$,若存在一个$\alpha_i^t(x)$较其他$\alpha_k^t(x)(k \in (1,\cdots,i-1,i+1,\cdots,m))$小很多,说明该子群内的个体或是找到最优解,或是陷入局部最优,则可以将该子群的个体分散到其他子群中,即$m_1 = m - 1$. 针对寻优过程中可能存在各个子群内的$\alpha_i^t(x)$相差不大的情况,为了加速迭代需计算$m_2 = \lfloor m - (m-1)\sin(\pi t/2T) \rfloor$,其中,$\lfloor\ \rfloor$为取整函数,$m$为初始的分组个数,迭代后期分组数设置为1. 最后取$m = \min\{m_1, m_2\}$.

这种分组方式使得在进化初期萤火虫种群以子群方式在各自的领域活动;随着不断进化其子群个数逐渐减少,已找到全局最优解的子群个体分散到其他子群有利于加速迭代速度,对于陷入局部最优的子群则有利于其中的个体跳出局部最优解. 由上面的分组方式可知,在迭代后期所有个体会合并为一个种群进行寻优.

2.3 分组内个体进化方式

2.3.1 对子群内的最优萤火虫个体的更新方式进行改进

在基本萤火虫算法中,最优个体通过随机移动位置来更新,若找到比当前位置更优的位置则代替,否则保持原个体. 虽然可以保持个体的多样性,但同时也导致个体移动的盲目性. 社会学研究表明:优秀个体的周围容易找到比其更优的个体,即局部最优值周围容易找到更优值. 采用云模型表示知识时,具备稳定中有变化、不确定性中存在确定性的特征,反映了自然界生物进化的基本原理[17-18]. 利用正态云模型算法对最优个体进行更新,将其视作一个云滴C(Ex,En,He),将最优个体作为期望(Ex)表示寻优中心;将本次迭代计算的适应度方差σ^2当作熵(En)来动态调整寻优范围;用超熵(He)来控制h个云滴的离散度,迭代初期加大随机性,迭代后期加强稳定性,令$\mathrm{He} = \mathrm{En}/\lambda$,$\lambda = \lambda_{\max} - (\lambda_{\max} - \lambda_{\min})t/T$,其中$\lambda_{\max}$和$\lambda_{\min}$为$\lambda$的取值区间($\lambda \in [3,10]$),$t$为当前迭代次

数,T为总的迭代次数. 若存在比原个体更优的个体,则取代原个体.

2.3.2　对子群内的非最优个体的更新方式进行改进

基本萤火虫算法的进化机理使其在求解高维连续优化问题时效果不够理想,在处理大规模复杂问题时容易陷入局部最优. 这是由于这些个体在更新时总是依赖于自己搜索范围内比自己更亮的萤火虫,虽然可以通过$\alpha\times(\mathrm{rand}()-1/2)$来避免陷入局部最优,但这种扰动缺少方向性. 最优觅食理论表明: 为获取更好的觅食效果,动物更趋向在觅食过程中以消耗更低的能效耗费来获得更多的猎物[19]. 以求最小值为例,利用下式计算两个萤火虫直接的能效吸引力:

$$F_{ik}=\frac{f(x_i)-f(x_k)}{r_{ik}}. \tag{4}$$

其中: F_{ik}为个体i对个体k的吸引力,仅考虑$F_{ik}>0$; $f(x_i)$和$f(x_k)$为个体的适应度; r_{ik}为萤火虫i与k之间的空间距离(与基本FA的相同,无需再计算). 设对个体x_i产生最大F_{ik}的萤火虫个体为x_k, x_j是子群内吸引度最大的个体,依据最优觅食理论改进个体更新方式,则第t代个体更新公式如下:

$$x_i'=x_i+\beta(x_i-x_j)+w(x_i-x_k)+\alpha\Big(\mathrm{rand}()-\frac{1}{2}\Big). \tag{5}$$

其中: rand()为[0,1]的随机数, $w\in[0,1]$为吸引系数. 由式(5)可知,吸引系数w的取值会影响寻优性能,在算法进化初期种群的多样性较多,这时适应度较差的个体可能离最优个体较远,其应该在向最优个体靠近时先获取足够的能量,此时w的取值应该较大; 而在进化后期多数个体已经趋于最优值的附近,此时w的取值应该较小,加快收敛速度. 以求最小值为例,给出w的自适应取值公式如下:

$$w=1-\frac{f_{\mathrm{best}}}{f_k}\sin\Big(\frac{\pi t}{2T}\Big). \tag{6}$$

2.3.3　最后一个分组的进化方式改进

由3.2的自适应分组策略可知,亮度相对小的萤火虫个体总是被划分到靠后的分组,致使该分组的寻优效果不如较靠前分组. 针对存在的这种局限性,可以采用差分进化算法更新最后分组的个体,用混沌理论完成个体变异. 差分进化算法[20]已被证明是进化算法中简单而最高效的算法[21],具有原理简单、受控参数少的优势. 其主要的变异操作在很大程度上影响着该算法的性能[22]. 采用下式更新组内个体:

$$x_i'=x_b+F(x_{r1}-x_{r2}). \tag{7}$$

其中: 随机选择互不相同且不同于该更新个体的整数,然后对适应度进行排序,选择3个个体最优的个体作为x_b,较优个体作为r_1,最差个体作为r_2, F为比例缩放因子. 如果适应度相差较小,则说明两个个体在空间中相隔很近, F_i应取较大值,以防止扰动量过小; 如果适应度相差较大,则说明两个个体在空间相隔很远, F_i应取较小值,以限制扰动量过大. 下面给出一种确定方式:

$$F_i=F_l+(F_u-F_l)\frac{f_{r1}-f_b}{f_{r2}-f_b}. \tag{8}$$

其中: F_u, F_l为F_i的上限和下限; f_{r1}, f_{r2}, f_b分别为个体r_1、r_2和x_b的适应度.

Rahnamaya证明了反向学习算法具有较快的学习速度和更强的优化能力[23]. 混沌映射产生的个体呈现遍历性、随机性和多样性[24],可有效地在收敛区域以外空间搜索全局最优位置. 文献[25]研究表明,偶数阶的Chebyshev映射可以生成随机性较好的个体. 针对该分组内连续迭代3次没有改善的萤火虫,按下式计算新个体:

$$\begin{cases} x'=a+b-x, \\ x_{\mathrm{new}}=\cos(\arccos x),\ k=2, \\ x'_{\mathrm{new}}=\cos(\arccos x'),\ k=2. \end{cases} \tag{9}$$

2.4　基于均匀设计的β_0、γ和α的自适应调整

将式(2)和(3)进行合并,得到萤火虫个体的更新公式如下:

$$x_i=x_i+\beta_0\mathrm{e}^{-\gamma {r_{ij}}^2}(x_j-x_i)+\alpha\Big(\mathrm{rand}()-\frac{1}{2}\Big). \tag{10}$$

由式(10)可知, β_0、γ和α在萤火虫算法的搜索过程中起着关键作用: β_0越大表明光越亮的萤火虫对周围萤火虫的吸引度越高,当$\beta_0=0$时,说明萤火虫个体在自身附近作随机运动. γ较小或是α较大有利于全局寻优,却影响了寻优精度,易于在寻优后期产生震荡; γ较大或是α较小虽有利于加强局部精细探索,但却降低了算法的收敛速度. 文献[1]给出了3个参数的取值范围: $\beta_0=[0,1]$, $\gamma=[0,10]$, $\alpha=[0,1]$. 已有学者[13,26]跟随迭代次数递减或递增相应的参数. 这种单纯的递增和递减不利于三者之间的协调,即他们之间不存在相互独立的增减,而是存在一定联系,进而影响着算法的寻优性能.

均匀设计[27]考虑试验点的均匀分布,能用较少的试验点获得最好的均匀性. 通过较少试验次数获得运行参数的最优配置,使算法可以以较优的状态运行. 文献[27]在算法每一次迭代过程中利用均匀设计法来动态调整β_0、γ和α,方法如下.

Step 1: 选取前$m-1$代获得最优萤火虫个体的运行参数β_0、γ和α.

Step 2: 按照下式确定本次迭代的运行参数($\beta_0 \in [\beta_{\min}, 1], \gamma \in [\gamma_{\min}, \gamma_{\max}]$和$\alpha \in [\alpha_{\min}, 1]$):

$$\begin{cases} \beta_0 = \beta_{\min} + \cos\left(\dfrac{\pi t}{2T}\right), \\ \alpha = \alpha_{\min} + \cos\left(\dfrac{\pi t}{2T}\right), \\ \gamma = \gamma_{\min} + \sin\left(\dfrac{\pi t}{2T}\right)(\gamma_{\max} - \gamma_{\min}). \end{cases} \tag{11}$$

Step 3: 确定均匀设计表$U_l(l^3)$,按照l个水平和3个属性编制实验方案.

Step 4: 根据实验方案的参数设置,按照锦标赛的方法选取n个个体进行进化,挑选一个能使$n/2$个以上的个体适应度都得到提高或是找到比当前最优个体还好的个体所对应的β_0、γ和α,将其作为当前代的计算参数.

Step 5: 若没有找到最佳组合,则利用Step 2的参数进行迭代计算.

2.5 算法流程

Step 1: 参数初始化,设置迭代次数$t = 1$,最大迭代次数T,子群内迭代次数limit和误差精度;

Step 2: 采用佳点集理论初始化萤火虫个体;

Step 3: 采用2.2节的方法完成萤火虫群的分组,每个子群分别执行limit次Step 4 ~ Step 6;

Step 4: 采用2.3节的方法完成个体寻优行为;

Step 5: 采用2.4节的方法完成光强吸收系数和吸引度的自适应调整;

Step 6: $t = t + 1$,若达到最大迭代次数或满足误差精度,则退出,否则跳转到Step 3继续迭代计算.

3 实验仿真

通过8个求解函数极值优化的例子来验证算法的优化性能,将所提出算法与粒子群算法(PSO)、遗传算法(GA)、混洗蛙跳算法(SFLA)、经典萤火虫算法(FA)和文献[13]算法(EOLFA)进行性能比较. 种群大小都设置为100,最大迭代次数设置为3 000,误差精度设置为10 e-10,对优化函数独立运行20次. FA参数设置[1] $\gamma = 1.0, \beta_0 = 1.0, \alpha = 0.2$; GA参数设置[1]: 交叉概率$P_c = 0.95$,变异概率$P_m = 0.05$; PSO参数设置[28]: 惯性因子为0.729 8,自身因子为1.496 18,全局因子为1.496 18; SFLA参数设置: 分为10个子群,每个子群10个青蛙,子群内部迭代次数为15次; AGD-FA的参数设置: $\gamma \in [1, 10], \beta_0 \in [0.02, 1], \alpha \in [0.2, 1]$,均匀设计水平数$l = 7$,子群个数从10递减到1,子群内迭代次数15次. 对比6种优化算法获得的最优结果、最差结果、平均结果、平均运行时间和方差,其中最优结果、最差结果反映解的质量,平均结果显示算法所能达到的精度,平均时间反映算法的收敛速度,方差反映算法的稳定性和鲁棒性.

$$f_1 = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, \quad -100 \leqslant x_i \leqslant 100; \tag{12}$$

$$f_2 = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10], \quad |x_i| \leqslant 5.12, n = 10; \tag{13}$$

$$f_3 = \sum_{i=1}^{n}|x_i| + \prod_{i=1}^{n}|x_i|, \ |x_i| \leqslant 10, \ n = 10; \tag{14}$$

$$f_4 = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3, \quad -100 \leqslant x_i \leqslant 100; \tag{15}$$

$$f_5 = \sum_{i=1}^{n} x_i^2, \ -100 \leqslant x_i \leqslant 100, n = 10; \tag{16}$$

$$f_6 = 20 + \exp(1) - 20\exp\Big[-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\Big] - \exp\Big[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\Big], \quad -32.768 \leqslant x_i \leqslant 32.768, n = 10; \tag{17}$$

$$f_7 = \frac{1}{4\,000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -600 \leqslant x_i \leqslant 600, n = 10; \tag{18}$$

$$f_8 = 100 + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)], \quad -5.12 \leqslant x_i \leqslant 5.12, n = 10. \tag{19}$$

由实验对比结果可知,文中提出的自适应分组差分萤火虫算法具有较好的求解速度和精度,可从以下3个方面进行分析.

1) 由表1~表8中的最优结果、最差结果、平均结果可以看出,AGD-FA寻优结果最好,主要因为采用佳点集理论和云模型算法加大了萤火虫个体靠近最优个体的几率,通过云模型的稳定倾向性可使局部最优个体较好地向其附近更优个体进行自适应定位,云模型的随机性又利于保护萤火虫个体的多样性,进而提高算法的寻优速率和性能. 而且加入的差分变异混沌算子有利于在迭代后期较好地防止算法陷入局部最优,有效地改善算法在求解一些高维复杂函数时求解精度不高、收敛速率慢和易于陷入局部极值的缺点.

2) 由运行时间和方差可以看出,AGD-FA的运行时间和方差都优于前几种算法,分析其原因,SFLA的进化方式较FA、PSO和GA相对简单,且调整参数少,同时利用小生镜的方式进行寻优,速度相对较快. SFLA在分组内迭代一定次数后再重新分组,在一定程度上与AGD-FA经过limit次迭代后进行自适

表 1　函数 f_1 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	1.209e-12	0.0091	8.93e-04	21.523	3.36e-01
GA	2.859e-10	0.627	3.62e-02	25.729	2.05e-02
SFLA	7.803e-12	8.507e-11	5.631e-11	19.305	3.853e-21
FA	3.618e-11	0.0805	4.62e-03	22.308	5.74e-02
EOLFA	8.083e-12	5.257e-8	2.164e-9	11.508	6.25e-15
AGD-FA	8.203e-13	9.215e-12	4.708e-12	7.154	4.923e-23

表 2　函数 f_2 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	32.892	145.614	102.562	22.891	3.17e-01
GA	39.805	164.206	136.132	25.095	1.07e+01
SFLA	2.109e-4	8.983e-2	6.018e-3	16.169	7.18e-05
FA	3.816	7.318	6.605	17.159	6.02e-01
EOLFA	6.135e-2	8.247e-1	8.013e-2	13.267	3.92e-03
AGD-FA	6.518e-10	7.412e-8	4.507e-9	7.372	4.95e-17

表 3　函数 f_3 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	3.582e-4	0.721	0.285	32.254	1.13e-02
GA	5.157e-2	6.93	4.531	35.519	2.39e+00
SFLA	-0.999	7.317	3.126	25.096	1.00e+00
FA	3.837e-2	1.237	3.639	21.064	2.03e+00
EOLFA	2.495e-10	6.476-7	5.498e-08	10.207	5.18e-15
AGD-FA	3.257e-12	7.329e-9	5.394e-10	6.534	4.31e-19

表 4　函数 f_4 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	2.237e-10	9.318e-8	5.674e-9	33.397	2.47e-13
GA	2.361e-8	7.842e-7	6.673e-8	35.635	6.21e-11
SFLA	3.172e-11	6.328e-9	7.256e-10	19.257	7.41e-14
FA	3.349e-9	5.925e-6	6.018e-7	23.017	6.89e-13
EOLFA	4.163e-11	8.027e-8	3.147e-9	12.249	5.09e-15
AGD-FA	2.205e-12	6.095e-9	4.209e-10	8.815	4.51e-19

应分组相似,但AGD-FA的分组个数随着迭代过程逐渐减少,有利于前期进行全局寻优,后期进行局部精细挖掘;且AGD-FA在进化过程中采用云模型和差分变异策略使得个体更新时既保持了随机性,又使得个体变化带有确定性,与经典算法的随机方式相比更易向最优解方向靠近;而基于均匀设计理论的参数调整有利于避免算法参数试算,虽然在每次迭代进化过程中增加了计算量,但由迭代次数可以看出,其收敛速度确实得到很大提高,使得运行时间相对较短.

3) 分析AGD-FA的算法原理可知,在不考虑种群初始化的时间复杂度情况下,改进算法在自适应分组策略和分组内个体进化方式增加了计算量,但由于经典FA在实现过程中已完成对亮度(适应度)的排序和个体间距离的计算,AGD-FA实际的计算开销主要增加在2.3.1节和2.3.3节更新方式上,针对2.3.1节中的方式,本质是利用云模型进行若干启发式计算,相当于增加若干个体;而式(2)与(8)的计算量相似,实际

表 5　函数 f_5 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	9.578e-11	7.347e-03	6.523e-05	27.387	1.22e-06
GA	2.761e-02	6.190e-01	2.325e-01	30.517	2.43e-01
SFLA	1.176e-10	9.899e-10	7.618e-10	14.59	6.61e-19
FA	1.827e-02	6.157e-02	3.302e-02	18.03	3.08e-01
EOLFA	2.209e-12	8.719e-10	7.259e-11	12.19	3.25e-21
AGD-FA	2.801e-13	8.216e-12	3.077e-12	7.012	1.29e-23

表 6　函数 f_6 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	8.398e-05	1.327	0.729	30.615	3.95e-02
GA	1.385e-01	5.097	2.326	32.171	1.42e+01
SFLA	1.805	2.015	1.862	16.878	3.56e+00
FA	5.016e-02	2.018e-01	3.129e-02	17.029	4.65e-01
EOLFA	3.098e-12	8.024e-11	7.625e-12	11.45	6.54e-21
AGD-FA	3.835e-13	9.016e-12	2.461e-12	7.473	4.35e-23

表 7　函数 f_7 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	2.163e-02	4.173e-01	1.609e-01	35.318	3.42e-02
GA	7.162e-02	6.021e-01	3.271e-01	38.035	1.12e-01
SFLA	3.148e-02	0.142	0.075	15.156	5.34e-03
FA	5.041e-02	6.325e-01	3.172e-01	17.524	3.18e-01
EOLFA	5.012e-13	7.023e-12	6.57e-12	8.24	3.26e-21
AGD-FA	7.015e-14	8.257e-13	5.981e-13	6.157	3.11e-23

表 8　函数 f_8 的运行仿真结果对比

算法	最优结果	最差结果	平均结果	平均时间/s	方差
PSO	1.305e-05	3.092	0.628	30.615	2.16e-01
GA	2.246e-03	4.617	0.715	33.412	4.63e-01
SFLA	4.380e-11	2.015	0.603	13.389	9.84e-02
FA	1.917e-04	3.823	0.698	31.658	3.28e-01
EOLFA	4.423e-10	7.206e-08	6.019e-09	20.761	7.16e-17
AGD-FA	7.157e-13	8.936e-12	6.035e-12	8.968	4.67e-23

是变异操作增加了计算量,但这也只是极小部分个体,由算法复杂性渐近理论可知,改进后的AGD-FA没有增加经典FA算法的复杂度.

4　结　论

针对萤火虫算法存在早熟、收敛速度慢且求解精度不高的缺点,本文提出一种自适应分组差分萤火虫算法,并将云模型算法、差分变异算法和混沌算法引入个体更新过程中,利用均匀设计表来自适应调整运行参数. 实验仿真结果验证了算法的良好性能,表现出较快的收敛速率和较好的全局寻优能力. 所提出算法为进化计算的研究进行了新的探索和尝试.

参考文献(References)

[1] Yang X S.Nature-inspired metaheuristic algorithms[M]. London: Luniver Press, 2008: 83-96.

[2] 于宏涛,高立群,韩希昌.求解旅行商问题的离散人工萤火虫算法[J]. 华南理工大学学报: 自然科学版, 2015, 43(1): 126-131.
(Yu H T, Gao L Q, Han X C. Discrete artificial firefly algorithm for solving traveling salesman problems[J]. J of

South China University of Technology: Natural Science Edition, 2015, 43(1): 126-131.)

[3] 田梦楚,薄煜明,陈志敏.萤火虫算法智能优化粒子滤波[J]. 自动化学报, 2016, 42(1): 89-96.
(Tian M C, Bo Y M, Chen Z M. Firefly algorithm intelligence optimized particle filter [J]. Acta Automatica Sinica, 2016, 42(1): 89-96.)

[4] Horng M H, Liou R J. Multilevel minimum cross entropy threshold selection based on the firefly algorithm[J]. Expert Systems with Applications, 2011, 38(12): 14805-14811.

[5] 王改革,郭立红,段红.基于萤火虫算法优化BP神经网络的目标威胁估计[J]. 吉林大学学报:工学版, 2013, 43(4): 1064-1069.
(Wang G G, Guo L H, Duan H. Target threat assessment using glowworm swarm optimization and BP neural network[J]. J of Jilin University: Engineering and Technology Edition, 2013, 43(4): 1064-1069.)

[6] Filter I, JrFister I, Yang X S, et al. A comprehensive review of firefly algorithms[J]. Swarm and Evolutionary Computation, 2013, 13(1): 34-46.

[7] Yang X S. Multiobjective firefly algorithm for continuous optimization[J]. Engineering with Computers, 2013, 29(2): 175-184.

[8] Gandomi A H, Yang X S, Talatahari S, et al. Firefly algorithm with chaos[J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(1): 89-98.

[9] Farahani S M, Abshouri A A, Nasiri B, et al. A Gaussian firefly algorithm[J]. Int J of Machine Learning and Computing, 2011, 1(5): 448-453.

[10] 周永权,黄正新,刘洪霞.求解TSP问题的离散型萤火虫群优化算法[J]. 电子学报, 2012, 40(6): 1164-1170.
(Zhou Y Q, Huang Z X, Liu H X. Dis Crete glowworm swarm optimization algorithm for TSP problem[J]. Acta Electronica Sinica, 2012, 40(6): 1164-1170.)

[11] Arora S, Singh S. The firefly optimization algorithm: convergence analysis and parameter selection[J]. Int J of Computer Applications, 2013, 69(3): 48-52.

[12] 王铭波,符强,童楠.基于模拟退火机制的多种群萤火虫算法[J]. 计算机应用, 2015, 35(3): 691-695.
(Wang M B, Fu Q, Tong N. Multi-group firefly algorithm based on simulated annealing mechanism[J]. J of Computer Applications, 2015, 35(3): 691-695.)

[13] 周凌云,丁立新,何进荣.精英正交学习萤火虫算法[J]. 计算机科学, 2015, 42(10): 211-216.
(Zhou L Y, Ding L X, He J R. Elite orthogonal learning firefly algorithm[J]. Computer Science, 2015, 42(10): 211-216.)

[14] 华罗庚, 王元. 数论在近代分析中的应用[M]. 北京: 科学出版社, 1978: 1-99.
(Hua L G, Wang Y. Applications of number-theoretic methods in approximate analysis[M]. Beijing: Science Press, 1978: 1-99.)

[15] 刘香品,宣士斌,刘峰.引入佳点集和猴群翻过程的人工蜂群算法[J]. 模式识别与人工智能, 2015, 28(1): 80-89.
(Liu X P, Xuan S B，Liu F. Artificial bee colony algorithm with good point set and turn process of monkey algorithm[J]．Pattern Recognition and Artificial Intelligence, 2015, 28(1): 80-89.)

[16] 毕晓君, 张磊. 基于混合策略的双种群约束优化算法[J]. 控制与决策, 2015, 30(4): 715-720.
(Bi X J, Zhang L. Dual population constrained optimization algorithm with hybird strategy[J]. Control and Decision, 2015, 30(4): 715-720.)

[17] 付斌, 李道国, 王慕快. 云模型研究的回顾与展望[J]. 计算机应用研究, 2011, 28(2): 420-426.
(Fu B, Li D G, Wang M K. Review and prospect on research of cloud model[J]. Application Research of Computers, 2011, 28(2): 420-426.)

[18] 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): 1082-1090.
(Zhang G W, He R, Liu Y, et al. An evolutionary algorithm based on cloud model[J]. Chinese J of Computers, 2008, 31(7): 1082-1090.)

[19] 崔志华, 曾建潮．微粒群优化算法[M]. 北京: 科学出版社, 2011: 65-66.
(Cui Z H, Zeng J C. Particle swarm optimization[M]. Beijing: Science Press, 2011: 65-66.)

[20] Rainer S, Price K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces[J]. J of Global Optimization, 1997, 11(4): 341-359.

[21] 李章维,周晓根,张贵军.一种动态自适应差分进化算法[J].计算机科学, 2015, 42(6): 52-56.
(Li Z W, Zhou X G, Zhang G J. Dynamic adaptive differential evolution algorithm[J]. Computer Science, 2015, 42(6): 52-56.)

[22] 孔祥勇,高立群,欧阳海滨,等.求解大规模可靠性问题的改进差分进化算法[J].东北大学学报:自然科学版, 2014, 35(3): 328-332.
(Kong X Y, Gao L Q, Ouyang H B, et a. An improved differential evolution algorithm for large scaleReliability problems[J]. J of Northeastern University: Natural Science, 2014, 35(3): 328-332.)

[23] 王燕.反向粒子群算法理论及及其应用研究[D].西安:西安工程大学理学院. 2011.
(Wang Y. The research of opposition-based particle swarm optimization and its application[D]. Xi'an: College of Science, Xi'an Polytechnic University, 2011.)

[24] 胥小波, 郑康锋, 李丹, 等．新的混沌粒子群优化算法[J]．通信学报, 2012, 33(1): 24-37.
(Xu X B, Zheng K F, Li D, et al. New chaos-particle swarm optimization algorithm[J]．J on Communications, 2012, 33(1): 24-37.)

[25] 刘金梅, 屈强. 几类混沌序列的随机性测试[J]．计算机工程与应用, 2011, 47(5): 46-49.
(Liu J M, Qu Q. Randomness tests of several chaotic sequences[J]. Computer Engineering and Applications, 2011, 47(5): 46-49.)

[26] 李瑞青.改进的萤火虫算法及应用[D].长春: 吉林大学软件学院, 2015.
(Li R Q. Improved firefly algorithm and application[D]. Changchun: College of Software, Jilin University, 2015.)

[27] 方开泰,马长兴.正交与均匀试验设计[M].北京:科学出版社, 2001: 3-34.
(Fang K T, Ma C X. Orthogonal and uniform experimental design[M]. Beijing: Science Press, 2001: 3-34.)

[28] Poli R, Kennedy J, Blackwell T. Particle swarm optimization[J]. Swarm Intelligence, 2007, 1(1): 33-57.

（责任编辑：孙艺红）