

带有引力搜索算子的烟花算法

朱启兵, 王震宇, 黄敏

(江南大学 轻工过程先进控制教育部重点实验室, 江苏 无锡 214122)

摘要: 针对烟花算法(FA)寻优过程中粒子间信息交流少、对最优点位置不在原点和原点附近的目标函数求解能力差的缺点, 提出带有引力搜索算子的烟花算法(FAGSO). 算子利用粒子间相互引力作用对粒子维度信息进行改善, 以提高算法的优化性能. 6个标准和增加位置偏移测试函数的仿真结果表明, FAGSO相比于FA、粒子群算法和引力搜索算法, 在寻优速度和寻优精度方面有更好的优化性能.

关键词: 烟花算法; 引力搜索; 偏移函数; 函数优化; 全局寻优

中图分类号: TP301

文献标志码: A

Fireworks algorithm with gravitational search operator

ZHU Qi-bing, WANG Zhen-yu, HUANG Min

(Key Laboratory of Advanced Process Control for Light Industry of Ministry of Education, Jiangnan University, Wuxi 214122, China. Correspondent: ZHU Qi-bing, E-mail: zhuqib@163.com)

Abstract: For the problems that the individuals including fireworks and sparks are not well-informed in the process of searching optimum, and the algorithm yields a poor result when being applied on shifted functions whose optimum are not at the origin or near the origin, a hybrid fireworks algorithm with the gravitational search operator(FAGSO) is proposed. The operator improves the particles dimension information through the gravity between individuals. Simulation experiments are conducted on 6 standard and shifted benchmark functions. Results show that the hybrid algorithm displays better performance compared to the fireworks algorithm(FA), the particle swarm optimization(PSO) algorithm and the gravitational search algorithm(GSA).

Keywords: fireworks algorithm; gravitational search; shifted function; function optimization; global optimization

0 引言

元启发式算法一直是优化算法研究领域中的一个热点, 它们在解决复杂的最优化问题上体现了良好的高效性和稳定性. 特别是基于群体智能的元启发式算法, 如粒子群算法^[1]、人工蜂群算法^[2]、蚁群算法^[3]. 此类算法利用群体粒子间信息和粒子与环境间信息, 在搜索域内寻找全局最优解以解决复杂困难的优化问题, 已广泛应用于各个工程应用领域.

通过模拟烟花爆炸中炸点扩散机制, Tan等^[4]提出了一种新颖的寻优搜索算法——烟花算法(FA). Bureerat等^[5]通过对标准函数的检测实验, 比较了12种不同优化算法的优化性能, 其中烟花算法表现良好, 优化结果明显好于粒子群算法和遗传算法. 研究学者对烟花算法进行了改进研究. Gao等^[6]提出了一种文化烟花算法, 通过提取烟花粒子的有利信息引导搜

索; Zheng等^[7]提出了一种带有差分进化算子的混合烟花算法, 通过变异、交叉过程增加粒子多样性; 曹炬等^[8]结合遗传算法思想, 提出了带有遗传算子的烟花算法. 现有对烟火算法的改进研究主要是针对最优点位置在原点或原点附近位置的检测函数进行的, 此时烟花算法展现了良好的优化性能, 但当把最优点位置不在原点或原点附近位置的函数作为检测函数时, 烟花算法的求解结果会变得很差^[9], 这大大影响了烟花算法的实际应用范围.

为了克服上述烟花算法中存在的问题, 本文结合引力搜索算法思想提出一种新的解决优化问题方法, 即带有引力搜索算子的烟花算法(FAGSO). 在标准的烟花算法中加入引力搜索算子, 通过粒子间相互引力作用与优越粒子集合进行信息交互, 产生新的火花粒子, 增加火花粒子的多样性, 提高算法的搜索精度和

收稿日期: 2015-10-20; **修回日期:** 2016-03-08.

基金项目: 国家自然科学基金项目(61271384, 61275155); 中央高校基本科研业务费专项基金项目(JUSRP51510).

作者简介: 朱启兵(1973—), 男, 教授, 博士, 从事农业信息感知与处理、智能优化算法等研究; 王震宇(1992—), 男, 硕士生, 从事智能优化算法的研究.

收敛速度, 削弱标准烟花算法映射规则和高斯变异算子带来的不良影响. 通过对标准测试集函数和增加位置偏移函数的实验, 验证了所提出算法的有效性.

1 烟花算法

烟花算法是一种模拟烟花爆炸过程的全局优化算法, 烟花爆炸的过程可以看作粒子在自己领域内搜索解的过程. 烟花算法的工作过程与一般群体智能优化算法相似, 首先随机产生 N 个烟花作为初始化群体, 然后让群体中的每个烟花经历爆炸和高斯变异过程产生新的火花粒子, 并应用设定的映射规则保证新的火花粒子仍处于可行域内, 最后在保留最优个体的前提下, 按基于距离的选择策略从生成的所有个体中选择出余下的 $N - 1$ 个个体, 共同组成下一代初始烟花. 这样周而复始, 逐一迭代下去, 直到满足精度要求或达到设定的最大迭代次数.

1.1 烟花算法实现过程

设 $X(t) = [x_1, x_2, \dots, x_i, \dots, x_N]$ 为第 t 次迭代的初始烟花集合. 其中: N 为烟花个数; $x_i \in R^D$ 为第 i 个烟花在解空间中的位置信息, 其适应度值为 $f(x_i)$. 对每一个烟花粒子 x_i 进行爆炸操作, 产生个数为 s_i 的火花粒子集合 $Y_i(t) = [y_{i,1}, \dots, y_{i,j}, \dots, y_{i,s_i}]$, $y_{i,j} \in R^D$ 与 x_i 的维数相同, 火花粒子 $y_{i,j}$ 的生成过程可由下式表述:

$$y_{i,j} = x_i + A_i B \text{rand}(-1, 1), \quad 1 \leq i \leq N, 1 \leq j \leq s_i. \quad (1)$$

其中: B 为一个 $1 \times D$ 维的随机矩阵, 矩阵元素的取值为 0 或 1; A_i 为烟花粒子 x_i 的爆炸幅度, 计算公式为

$$A_i = \hat{A} \frac{f(x_i) - \min_{1 \leq k \leq N} f(x_k) + \varepsilon}{\sum_{k=1}^N (f(x_i) - \min_{1 \leq k \leq N} f(x_k)) + \varepsilon}, \quad (2)$$

\hat{A} 为一个常数以限制最大爆炸幅度, ε 为一个极小的常数, 以避免出现零的情况.

每个烟花 x_i 可产生的火花个数 s_i 由下式决定:

$$s_i = m \frac{\max_{1 \leq k \leq N} f(x_k) - f(x_i) + \varepsilon}{\sum_{i=1}^N (\max_{1 \leq k \leq N} f(x_k) - f(x_i)) + \varepsilon}, \quad (3)$$

其中 m 为常数, 用来限制产生的火花总数. 为避免爆炸产生火花数量过多或过少, 为每个烟花设定如下产生火花数量的限制公式:

$$\hat{s}_i = \begin{cases} \text{round}(am), & s_i < am; \\ \text{round}(bm), & s_i > bm, a < b < 1; \\ \text{round}(s_i), & \text{otherwise.} \end{cases} \quad (4)$$

其中: \hat{s}_i 为第 i 个烟花最终可产生的火花数量, $\text{round}()$ 为四舍五入取整函数, a 和 b 为给定的常数.

为进一步提高火花种群多样性, 烟花算法在求解过程中引入高斯变异过程, 从 $X(t)$ 集合中随机选择 p 个烟花 ($0 < p \leq N$), 对其按照式 (5) 进行高斯变异操作, 产生高斯变异火花粒子集合

$$Z(t) = [z_1, z_2, \dots, z_h, \dots, z_p],$$

有

$$z_h = x_h B g, \quad 1 \leq h \leq p. \quad (5)$$

其中: x_h 为从 $X(t)$ 中随机选择的烟花; B 为一个 $1 \times D$ 维的随机矩阵, 矩阵元素的取值为 0 或 1; g 为服从均值为 1、方差为 1 高斯分布的随机数.

为防止新产生的两种火花粒子超出搜索范围, 烟花算法采用模运算的映射规则, 将超出可行域范围外的火花拉回到可行域范围内. 如当火花粒子 $y_{i,j}$ 超出可行域范围时, 按下式计算:

$$y_{i,j}^d = y_{\min}^d + |y_{i,j}^d| \% (y_{\max}^d - y_{\min}^d). \quad (6)$$

其中: $y_{i,j}^d$ 为火花粒子 $y_{i,j}$ 在第 d 维上的位置, y_{\max}^d 和 y_{\min}^d 分别为第 d 维的上下搜索边界, $\%$ 为模运算.

每一次迭代过程的最后, 烟花算法从集合 $W(t) = \{X(t) \cup Y(t) \cup Z(t)\}$ 中选择 N 个粒子作为下一次迭代的初始烟花. 粒子群中最优个体会被保留下来, 其他 $N - 1$ 个烟花采用轮盘赌方式进行随机选择. 个体 $w_i \in W(t)$ 被选择概率基于粒子间距离进行计算, 其计算公式为

$$L(w_i) = \sum_{j=1}^{W(t)} \|w_i - w_j\|, \quad (7)$$

$$P(w_i) = \frac{L(w_i)}{\sum_{j \in W(t)} L(w_j)}. \quad (8)$$

其中: $L(w_i)$ 为个体 w_i 与其他个体的距离之和, 采用欧氏距离度量; $P(w_i)$ 为个体 w_i 被选择概率. 烟花算法流程如图 1 所示.

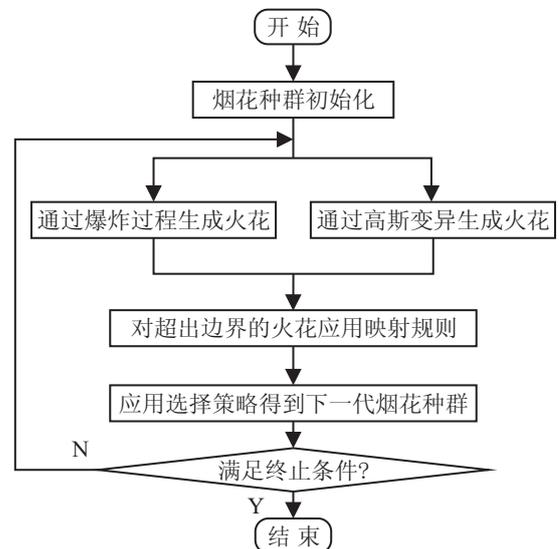


图 1 烟花算法流程

1.2 烟花算法分析

由式(1)~(3)可知, 对于适应度值好的烟花会在较小爆炸区域内产生更多的火花以加快收敛过程, 适应度值差的烟花会在较大爆炸区域内产生较少火花以增加搜索粒子的多样性. 但是, 这种粒子产生机制没有利用粒子群中其他优秀粒子的位置信息和整个群体信息, 粒子之间交流较少, 不能充分保证粒子的多样性且不够灵活. 同时研究发现, 烟花算法对于最优点位置处于原点或原点附近位置的测试函数有着十分明显的性能优势, 但当烟花算法应用到具有偏移的测试函数上进行求解, 即测试函数的最优位置不处于原点或其附近时, 标准烟花算法的求解结果会变的很差^[9]. 这主要由两个原因造成: 在烟花算法中, 对于超出边界的解, 映射规则很容易将其映射到搜索空间中间原点附近的位置, 如果测试函数的全局最优点在原点位置, 且搜索空间的上下边界对称, 则这种映射规则会使超出边界的点被人为设定到原点位置附近, 无意中加速了算法的收敛性; 如果一个解的位置已经十分接近原点位置, 则高斯变异算子很容易将解变异到原点位置附近, 使得该解很难跳出原点位置附近的区域.

2 带有引力搜索算子的烟花算法

对于一个 D 维优化问题而言, 求得优化解的好坏与每一维上的优化结果相关. 在搜索最优解的过程中, 较差适应度值的解很可能是因为在某些维度上没有取得最优的优化结果^[10]. 目前所研究的群体智能算法, 如差分进化算法、粒子群算法、引力搜索算法^[11], 都强调在求解最优化问题时充分利用粒子群的可利信息, 生成新粒子或更新原有粒子. 万有引力搜索算法的求解过程是通过粒子间万有引力的作用, 使得粒子朝着质量最大的粒子移动, 质量最大的粒子占据最优位置, 从而在移动的过程中得到优化问题的最优解^[12-13]. 结合以上思想和万有引力搜索算法的求解过程, 本文设计了引力搜索算子, 将算子功能加入到烟花算法求解优化问题过程中.

2.1 引力搜索算子

引力搜索算子的运算过程是通过粒子间相互引力作用对粒子进行位移操作, 进而改变粒子的维度信息, 其运动过程遵循动力学规律. 在此过程中, 粒子之间实现了优化信息的共享, 处于求解空间中的粒子能感应到全局环境信息并改善较差维度值, 引导粒子群体向最优解区域搜索. 算子具体操作过程如下:

设集合 $W(t) = \{X(t) \cup Y(t) \cup Z(t)\}$ 为 t 时刻, N 个烟花经爆炸、高斯变异后所得到的全部粒子集合, 粒子的总数为 N_a , 并设集合中的第 i 个粒子为

$$w_i \in R^D, \text{ 其惯性质量 } M_i \text{ 可计算为}$$

$$m_i = \frac{f(w_i) - \max_{w_j \in W(t)} f(w_j)}{\min_{w_j \in W(t)} f(w_j) - \max_{w_j \in W(t)} f(w_j)}, \quad (9)$$

$$M_i = \frac{m_i}{\sum_{w_j \in W(t)} m_j}. \quad (10)$$

由式(9)和(10)可以推导出, 对于单个粒子而言, 其适应度值越低质量越大, 既位置较优的粒子有更大的惯性质量. 算子从总数为 N_a 的粒子中选取质量最好(适应度小)的前 $2 \times N$ 个粒子组成优越粒子集合 R , 以集合 R 作为算子中的吸引单元, 对粒子集合 $W(t)$ 施加引力作用, 其引力为

$$F_i^d = \sum_{j \in R, j \neq i} \text{rand}(0, 1) F_{i,j}^d, \quad (11)$$

$$F_{i,j}^d = G \frac{M_i \times M_j}{r_{ij} + \varepsilon} (w_j^d - w_i^d). \quad (12)$$

其中: w_i^d 、 w_j^d 为集合 $W(t)$ 中第 i 、 j 粒子在第 d ($d \leq D$) 维上的坐标; F_i^d 为粒子 i 在第 d 维空间所受力总和; G 为引力常数, 其大小控制位置改变量的精度; M_i 为粒子的惯性质量; r_{ij} 为粒子 i 与粒子 j 之间的欧氏距离; ε 为一个极小常数, 防止分母为零.

在引力 F_i^d 的作用下, 粒子 w_i 在每一维上产生移动, 设移动后的粒子空间位置为 v_i , 其位置为

$$v_i^d = x_i^d + F_i^d / M_i. \quad (13)$$

记 $V(t) = [v_1, v_2, \dots, v_i, \dots, v_{N_a}]$ 为更新后的所有粒子位置, 根据粒子的适应度值大小, 从集合 $\{W(t) \cup V(t)\}$ 中选择 N 个烟花粒子作为下一次迭代过程的初始烟花, 记为 $X(t+1)$.

2.2 FAGSO 算法优化流程

将引力搜索算子加入到标准烟花算法中构成 FAGSO 算法, 算法优化求解流程如下.

Step 1: 设置算法参数 T 、 N 、 a 、 b 、 m 、 \hat{A} 、 p 、 G . 其中: T 为设定的迭代次数, N 为烟花粒子数目, a 、 b 、 m 为给定的常数以确定可生成火花粒子数目的最大最小值, \hat{A} 为限制烟花粒子爆炸幅度的常数, p 为要进行高斯变异的烟花个数, G 为引力常数.

Step 2: 在 n 维搜索空间内随机初始化 N 个烟花粒子的位置, 令 $t = 1$.

Step 3: 采用式(2)~(4)计算当次迭代过程中, 每个烟花产生的火花个数和爆炸幅度, 由式(1)对每个烟花执行爆炸操作, 产生普通火花粒子, 由式(5)对一定数量的烟花粒子进行高斯变异, 产生高斯变异火花粒子.

Step 4: 对超出范围的火花粒子按式(6)规则映射到可行域范围内.

Step 5: 由式 (9) 和 (10) 计算搜索空间中每个粒子的惯性质量, 并从中提取质量好的 $2 \times N$ 个粒子组成优越粒子集合 R .

Step 6: 计算集合 R 中优越粒子与其他每一个粒子的欧氏距离, 并由式 (11) 和 (12) 得到每个粒子每一维度受集合 R 的引力大小. 按式 (13) 产生新的火花粒子, 对超出范围的火花粒子按式 (6) 规则映射到可行域范围内. 将所有烟花和火花粒子按照适应度值大小排列, 从中选择出前 N 个粒子作为下一次迭代的初始烟花, 置 $t = t + 1$.

Step 7: 若 $t < T$, 则返回 Step 3, 否则算法停止, 输出目标函数的优化结果.

引入引力搜索算子后, 以适应度值好的 $2 \times N$ 个火花粒子作为引力单元, 使位置较差的火花粒子有机会与优越粒子在每一维度上进行信息交互, 产生位置信息改进后的火花粒子, 增加粒子的多样性以防止陷

入局部最优, 更加有利于寻找到全局最优. 对于增加位置偏移的测试函数, 即最优位置不在原点及原点附近的测试函数, 引力搜索算子的加入可削弱映射规则和高斯变异过程对优化过程所带来的不良影响.

3 实验分析

3.1 实验设计

为检验所提出算法的寻优精度、收敛速度等求解性能, 选用一组包含单峰和多峰的标准测试函数进行实验, 比较 FAGSO 算法与 FA 算法、粒子群算法 (PSO)、引力搜索算法 (GSA) 的优化性能. 其中: $f_1 \sim f_3$ 为单峰函数, 可以测试算法的寻优精度, 考察算法的执行性能; $f_4 \sim f_6$ 为多峰函数, 局部最优的数目会随着维数的增加呈指数增长, 用来检验算法跳出局部最优的能力. 表 1 给出 6 个检测函数的表达式、搜索范围、最优位置、理论最优值和搜索维度.

表 1 测试函数

函数名称	表达式	搜索范围	最优位置	最优值	搜索维度
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]$	$[0, 0, \dots, 0]$	0	30
Schwefel's 2.22	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i$	$[-10, 10]$	$[0, 0, \dots, 0]$	0	30
Rosenbrock	$f_3(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30, 30]$	$[0, 0, \dots, 0]$	0	30
Ackley	$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e\right)$	$[-32, 32]$	$[0, 0, \dots, 0]$	0	30
Griewank	$f_5(x) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]$	$[0, 0, \dots, 0]$	0	30
Rastrigin	$f_6(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	$[0, 0, \dots, 0]$	0	30

表 2 偏移指数与偏移量

SI	1	2	3	4	5	6
SV	$0.05 \times \frac{U_k - L_k}{2}$	$0.1 \times \frac{U_k - L_k}{2}$	$0.2 \times \frac{U_k - L_k}{2}$	$0.3 \times \frac{U_k - L_k}{2}$	$0.5 \times \frac{U_k - L_k}{2}$	$0.7 \times \frac{U_k - L_k}{2}$

对于每一个测试函数, 烟花种群初始化范围为 $[U_k/2, U_k]$, U_k 表示第 k 维的搜索范围上界. 实验用 Matlab 实现, FA 算法和 FAGSO 算法参数设定如下: 初始烟花数目 $N = 5$, 发生高斯变异烟花数目 $p = 5$, $m = 50$, $a = 0.04$, $b = 0.8$, 最大爆炸幅度 $\hat{A} = 40$, 引力搜索算子中设置 $G = G_0 \times e^{-\alpha}$, $G_0 = 100$, $\alpha = 20$, 此时对维度信息的改变量精度最高^[11]. PSO 算法和 GSA 算法参数的设定参考文献 [14, 11]. 实验设定每选择出一代初始烟花为一次迭代过程, 首先对标准被测函数进行实验, 设置迭代次数为 1000, 每个函数独立运行 20 次.

为检验 FAGSO 算法对偏移函数的优化能力, 对增加位置偏移量测试函数进行实验, 引用文献 [9] 中 6 种偏移指数和偏移量分析改进后的优化算法在不同偏移量下的优化性能. 位置偏移大小与优化问题的搜索范围相关, 如表 2 所示. U_k 、 L_k 分别为优化函数的搜索范围上下界, SI 为偏移指数, SV 为偏移量. 当偏移指数为 0 时, 被测函数未发生偏移; 当偏移指数不为 0 时, 粒子求适应度时将对应偏移量增加到粒子 x_i 的每一维位置信息上, 从而使被测函数的最优位置远离搜索区域原点处. 例如, 对于 f_1 -Sphere 函数, 当偏移指数为 6 时, 每一维上的改变量为 $0.7 \times$

$((100 - (-100))/2) = 70$, 最优点的位置偏移原点处 70, 搜索范围不发生改变. 偏移函数实验两种算法的参数设置与上相同, 设置迭代次数为 3 000, 每个函数独立运行 20 次.

3.2 实验结果和分析

3.2.1 标准函数实验结果和分析

表 3 为 6 种函数在无位置偏移情况下, 20 次独立运行获得的最优值和统计平均. 为了便于比较, 同时列出了有关文献中 PSO^[14]、GSA^[11]对这 6 种函数的优化结果. 由表 3 可见, 无论对于单峰函数还是多峰函数, FAGSO、FA 的整体性能都要优于 PSO 和 GSA, 且 FAGSO 的性能表现更加优异. 对于单峰函数 f_1 , FAGSO 在设定的迭代次数内求解得到了全局最优解. 对于 f_2 和 f_3 , FAGSO 也有最好的寻优精度. FA 虽然得到了相比 PSO、GSA 更好的结果, 但其收敛速度要比 FAGSO 慢, 收敛曲线比较如图 2(a) 所示. 对于多峰函数 f_4 、 f_5 、 f_6 , FAGSO 与 FA 在设定的迭代次数内得到了相同解, 且对于 f_5 和 f_6 都找到了全局最优解, 但 FAGSO 有更快的收敛速度, 收敛曲线比较如图 2(b) 所示. 以上结果表明, 引力搜索算子能有效地增加优秀粒子与其他较差粒子间的信息交互, 对较差粒子的位置信息进行改善, 在算法优化求解过程中起到了推动作用, 使算法有更快的收敛速度, 有利于求解到精度更好的最优解. 同时, 算子增加了粒子的多样性, 对于局部空间内有多个极小值点的优化问题有更好的求解性能.

表 3 FAGSO 与其他算法的比较

函数	FAGSO	FA	PSO	GSA
f_1	best	0	2.38e-229	4.39
	mean	0	1.96e-163	42.37
f_2	best	2.88e-215	8.67e-118	2.03
	mean	8.44e-192	3.47e-93	19.95
f_3	best	3.1e-3	0.67	1.47e+2
	mean	16.96	19.90	2.81e+3
f_4	best	8.88e-16	8.88e-16	5.29
	mean	8.88e-16	8.88e-16	10.13
f_5	best	0	0	1.03
	mean	0	0	1.55
f_6	best	0	0	39.50
	mean	0	0	84.85

3.2.2 偏移函数实验结果和分析

表 4 为不同偏移量情况下, FAGSO 和 FA 的实验结果. 由表 4 可见, 随着偏移量的增加, FAGSO 和 FA 的寻优精度在整体上都有所下降, 且 FA 的下降幅度尤为明显, 其原因在于, 随着偏移量的增加, FA 中高斯变异和映射规则已不再起关键作用. 在不同偏移量

下, FAGSO 相比于 FA 在寻优精度上有很大提升, 且 FAGSO 的标准差要远小于 FA, 这表明 FAGSO 相比 FA 更加稳定.

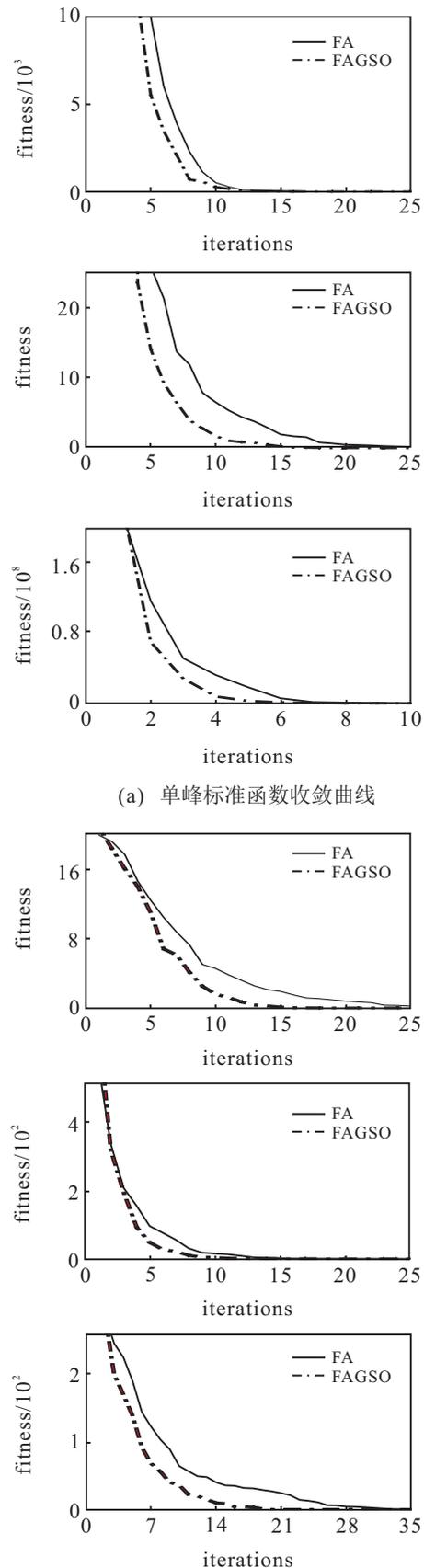
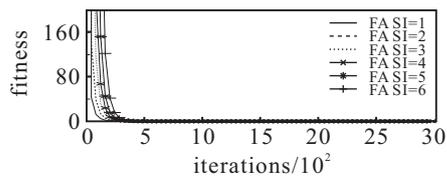
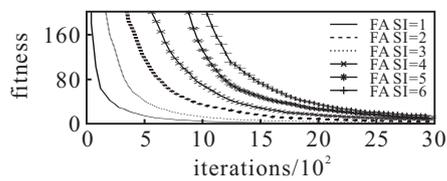


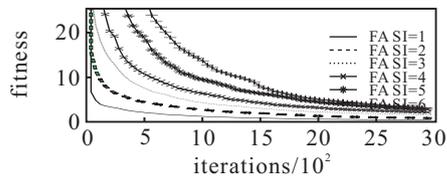
图 2 标准函数收敛曲线

表4 不同偏移指数下实验结果

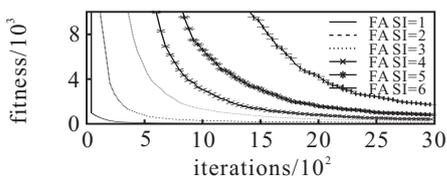
		f_1 -Sphere		f_2 -Schwefel's		f_3 -Rosenbrock		f_4 -Ackley		f_5 -Griewank		f_6 -Rastrigin	
		FA	FAGSO	FA	FAGSO	FA	FAGSO	FA	FAGSO	FA	FAGSO	FA	FAGSO
SI=1	平均值	6.26e-1	2.78e-9	5.28e-1	4.59e-2	3.99	7.30e-3	5.21e-1	7.33e-4	2.24e-1	1.68e-2	6.01	1.48
	标准差	2.21e-1	1.29e-9	1.97e-1	1.85e-2	1.61	1.25e-2	4.46e-1	2.56e-4	7.87e-2	2.50e-2	3.92	1.25
SI=2	平均值	1.60	3.62e-9	9.99e-1	5.98e-2	121.48	13.14	1.15	1.73e-3	4.30e-1	1.60e-2	10.36	2.39
	标准差	4.86e-1	2.41e-9	3.03e-1	2.12e-2	50.13	20.35	4.73e-1	2.60e-2	1.08e-1	2.10e-2	4.45	2.16
SI=3	平均值	3.39	3.35e-9	1.61	8.21e-2	254.88	29.88	2.02	2.35e-2	6.29e-1	7.51e-3	4.62	1.43
	标准差	8.33e-1	2.41e-9	4.97e-1	2.96e-2	104.56	25.77	3.54e-1	9.87e-2	1.10e-1	7.20e-3	2.28	1.20
SI=4	平均值	7.23	3.02e-9	2.22	1.04e-1	409.52	94.83	2.28	7.47e-2	6.77e-1	1.45e-2	13.07	2.28
	标准差	2.79	9.42e-9	5.51e-1	4.11e-2	180.43	297.67	4.82e-1	1.95e-1	1.22e-1	1.27e-2	3.85	1.53
SI=5	平均值	9.69	2.95e-9	2.77	1.02e-1	808.29	88.97	2.73	8.84e-2	8.24e-1	1.28e-2	15.07	2.27
	标准差	3.45	1.32e-9	6.17e-1	3.16e-2	490.93	122.97	3.52e-1	2.49e-1	1.19e-2	1.84e-2	3.10	1.20
SI=6	平均值	11.57	3.40e-9	3.12	9.93e-2	1638.89	180.26	3.19	7.40e-2	8.80e-1	1.32e-2	15.58	2.13
	标准差	5.80	1.53e-9	7.20e-1	4.15e-2	900.78	280.94	5.03e-1	1.84e-1	9.71e-2	1.38e-2	3.69	9.46e-1



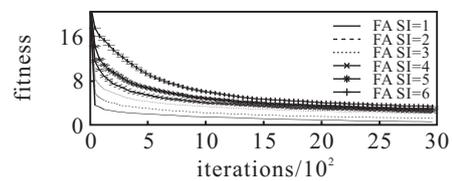
(a) f_1 -Sphere 收敛曲线



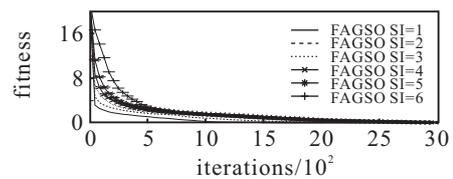
(b) f_2 -Schwefel 收敛曲线



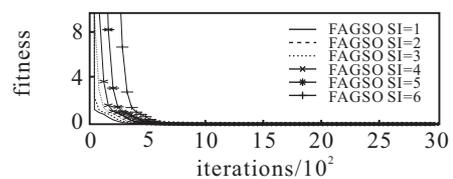
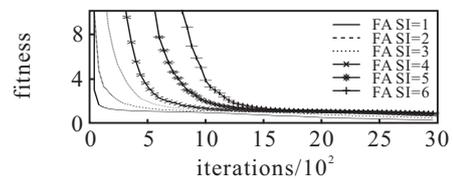
(c) f_3 -Rosenbrock 收敛曲线



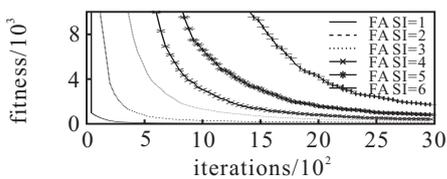
(a) f_4 -Ackley 收敛曲线



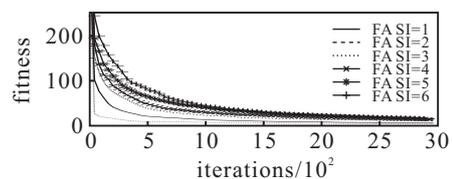
(b) f_5 -Griewank 收敛曲线



(c) f_6 -Rastrigin 收敛曲线



(c) f_3 -Rosenbrock 收敛曲线



(c) f_6 -Rastrigin 收敛曲线

图3 单峰偏移函数收敛曲线

图4 多峰偏移函数收敛曲线

图 3 和图 4 为不同偏移量下, 利用 FA 和 FAGSO 优化 f_1 、 f_2 、 f_3 单峰函数和 f_4 、 f_5 、 f_6 多峰函数的收敛曲线。由曲线可见, 随着偏移量的增加, FA 的收敛性能急剧下降, 在设定的迭代次数 (3 000 次) 内, 难以收敛到最优值, FAGSO 的收敛速度明显优于 FA。这表明, FAGSO 防止陷入局部最优的能力相比于 FA 有所加强, 更加有利于解决高维复杂的优化问题。

4 结 论

针对烟花算法求解过程中的弊端, 本文结合引力搜索算法思想提出了一种带有引力搜索算子的烟花算法。该算法在优化求解过程中利用优越火花粒子的位置信息, 对其他粒子位置信息进行改善, 增强粒子群间的信息交流, 削弱了烟花算法中映射规则和高斯变异过程对求解过程带来的不良影响, 提高了算法的优化性能。从对 6 个标准测试函数和偏移函数的实验结果看, 所提出算法不仅能得到较好的优化结果, 而且具有较高的收敛速度。

从研究结果可以发现, 尽管本文提出的算法可以改善偏移条件下的寻优精度, 但是其在迭代后期的收敛速度仍有进一步提高的空间。如何吸收其他群智能算法的优点, 提高其在偏移条件下迭代后期的收敛速度将是下一步需要解决的问题。

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization[J]. *Swarm Intelligence*, 2007, 1(1): 33-57.
- [2] Karaboga D, Bastuk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony(ABC) algorithm[J]. *J of Global Optimization*, 2007, 39(3): 459-471.
- [3] Dorigo M, Maniezzo V. Ant system: Optimization by a colony of cooperating agents[J]. *Systems, Man, and Cybernetics*, 1996, 26(1): 29-41.
- [4] Tan Y, Zhu Y. Fireworks algorithm for optimization[J]. *Lecture Notes in Computer Science*, 2010: 21(7): 355-364.
- [5] Bureerat S. Hybrid population-based incremental learning using real codes[C]. *Learning and Intelligent Optimization*. Rome: Springer-Verlag, 2011: 379-391.
- [6] Gao H, Diao M. Cultural fireworks algorithm and its application for digital filters design[J]. *Int J of Modelling, Identification and Control*, 2011, 14(4): 324-331.
- [7] Zhen Y J, Xu X L, Ling H F, et al. A Hybrid fireworks optimization method with differential evolution operators[J]. *Neurocomputing*, 2015, 148(148): 75-82.
- [8] 曹炬, 李婷婷, 贾红. 带有遗传算子的烟花爆炸优化算法[J]. *计算机工程*, 2010, 36(23): 149-151.
(Cao J, Li T T, Jia H. Fireworks explosion optimization algorithm with genetic operators[J]. *Computer Engineering*, 2010, 36(23): 149-151.)
- [9] Zheng S, Janecek A, Tan Y. Enhanced fireworks algorithm[C]. *Evolutionary Computation*. Cancun: IEEE, 2013: 2069-2077.
- [10] Liang J, Qin A, Suganthan P, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *IEEE Trans on Evolutionary Computation*, 2006, 10(3): 281-295.
- [11] Rashedi E, Nezamabadi H, Saryazdi S. GSA: A gravitational search algorithm[J]. *Information Science*, 2009, 179(13): 2232-2248.
- [12] Li C, Zhou J. Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm[J]. *Energy Conversion and Management*, 2011, 52(1): 374-381.
- [13] 徐遥, 王士同. 引力搜索算法的改进[J]. *计算机工程与应用*, 2011, 47(35): 188-192.
(Xu Y, Wang S T. Enhanced version of gravitational search algorithm: weighted GSA[J]. *Computer Engineering and Applications*, 2011, 47(35): 188-192.)
- [14] Zhao X. A perturbed particle swarm algorithm for numerical optimization[J]. *Applied Soft Computing*, 2010, 10(1): 119-124.

(责任编辑: 郑晓蕾)