

群活性反馈的变异自适应分数阶粒子群优化

苏守宝^{1,2}, 陈秋鑫^{1,2}, 王池社¹, 李智^{1,2}

(1. 金陵科技学院 数据科学与智慧软件江苏省重点实验室, 江苏南京 2111169;
2. 江苏科技大学 计算机学院, 江苏镇江 212003)

摘要: 针对传统分数阶粒子群优化(FOPSO)在算法综合性能上依赖于分数阶次 α ,易陷入早熟收敛,为此提出一种基于群活性反馈的S型自适应分数阶粒子群方法(SFOPSO),即根据种群活性以及粒子个体的活跃程度自适应动态调整每个粒子的分数阶次 α ,使种群在搜索过程中保持较好的稳定性与多样性;同时设计了一种混合变异机制以提升种群在探索期和开发期跳出局部最优的能力.理论分析证明了提出的算法SFOPSO的收敛性,实验选取6个不同特征的基准优化函数进行测试,结果证明了所提出SFOPSO算法的可行性和有效性,5种方法性能比较分析表明,SFOPSO具有更好的收敛精度和收敛速度.

关键词: 粒子群优化;自适应;变异算子;分数阶;群活性

中图分类号: TP18,TP303.6 **文献标识码:** A doi: 10.3969/j.issn.0253-2778.2020.07.021

引用格式: 苏守宝,陈秋鑫,王池社,等. 群活性反馈的变异自适应分数阶粒子群优化[J]. 中国科学技术大学学报,2020,50(7):1026-1034.

SU Shoubao, CHEN Qiuxin, WANG Chishe, et al. Adaptive fractional order particle swarm optimization using swarm activity feedback and mutation operator[J]. Journal of University of Science and Technology of China, 2020,50(7):1026-1034.

Adaptive fractional order particle swarm optimization using swarm activity feedback and mutation operator

SU Shoubao^{1,2}, CHEN Qiuxin^{1,2}, WANG Chishe¹, LI Zhi^{1,2}

(1. Jiangsu Key Laboratory of Data Science and Smart Software, Jinling Institute of Technology, Nanjing 211169, China;
2. School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: The basic particle swarm optimizer with fractional-order (FOPSO) is easy to fall into premature convergence, because its overall performance depends on the fractional order α . To solve the problem, a new adaptive fractional-order PSO algorithm, SFOPSO is proposed, by cooperating mutation operators into swarm activity feedback with S-model. During the iteration of this new algorithm, the fractional-order α of particles is adjusted adaptively according to the swarm activity with S-model and the activity value of single particles. At the same time, to enhance the ability of the swarm to escape out of local optimum during the process of exploitation or exploration, the hybrid model was designed by using mutation operators. The convergence of the proposed algorithm SFOPSO is analyzed theoretically and the experimental results show that the proposed algorithm is practicable and effective in improving convergence accuracy and convergence speed.

Key words: particle swarm optimization; adaptive, mutation operator; fractional-order; swarm activity

0 引言

粒子群优化 (particle swarm optimization, PSO)是最典型的群体智能计算模型之一,由于其具有结构简约、易于实现、无梯度信息限制等特点,在科学研究、工程应用等诸多方向有着独特的优势,因此该算法得到了快速发展并广泛应用于生物工程、工业设计、自动化控制、信号处理、聚类分析等

多个领域.传统 PSO 算法在寻优过程中,由于算法结构较为简单,算法的控制参数直接影响算法的精度、可靠性和收敛性,存在易陷入局部最优、早熟收敛和收敛性能差等问题^[1].为了解决这些问题,诸多学者通过调整参数^[2]、改变粒子速度更新方式^[3]、融合遗传算法“变异”思想^[4]等方案来提升算法的性能.

分数阶微积分理论 (fractional order calculus,

收稿日期: 2020-04-23; 修回日期: 2020-07-28

基金项目: 国家自然科学基金(61375121, 41801303), 金科院高层次引进人才科研项目(jit-rcyj-201505), 江苏省高校省级自然科学研究重大项目(17KJA520001, 18KJA520003), 江苏省高校优秀科技创新团队项目(苏教科[2017]6号)资助.

作者简介: 苏守宝,男,1965年生,博士/教授.研究方向:群智能大数据挖掘等. E-mail: showbo@jit.edu.cn

通讯作者: 王池社,博士/教授. E-mail: wangcs@jit.edu.cn

FOC)作为一种有效的自然科学和工程计算工具,在结构设计、预测分析、调度优化、数据挖掘、图像处理与模式识别等实际工程设计与优化领域中取得了成功应用^[5-9].由于 PSO 是模拟基于个体(individual)的生物种群经过较长时期不断进化过程的动态寻优算法,分数阶微分 FOC 因其固有的长期记忆等显著特性,非常适宜于描述 PSO 的搜索寻优过程,因此 Pires 等^[10]通过分数阶动力系统研究粒子群算法中粒子的寻优变化过程以及算法参数对全局动力的影响,将分数阶微积分 Grünwald-Letnikov 引入到粒子个体速度和位置更新计算公式,使用分数阶阶次(fractional-order, FO)以控制算法的收敛速度、搜索性能和精度.在此基础上,Pires 等^[11]利用分数阶微积分改进 PSO、设计出分数阶粒子群优化算法(FOPSO),对粒子的速度变化值求分数阶导数,得出了新的粒子速度更新方式.随后, Couceiro 等^[12-14]为了提高分数阶粒子群 FOPSO 的算法性能,将达尔文进化(DPSO)思想引入分数阶 PSO 优化,提出了基于 Darwinian 进化的混合分数阶粒子群方法(FODPSO);郭等^[15]分析了达尔文 PSO 的收敛性并实验测试了混合 PSO 的算法性能,以找出保证 FODPSO 综合性能较优的算法参数取值范围,实验表明 FODPSO 算法在大多数问题的求解中,收敛精度与速度相较 FOPSO 算法有一定的提升,但是分数阶次 α 依然是影响 FODPSO 算法收敛性能的重要因素.当 α 取值较大时,算法收敛速度较慢,而当 α 取值较小时,粒子群容易陷入局部最优导致算法早熟收敛.此外,DPSO 算法自身特性也给 FODPSO 算法增加了计算复杂度和不确定因素^[15-16].

针对上述问题,受文献[17-18]的启发,将群活性反馈策略与 Sigmoid 函数型非线性参数调整策略运用到 FOPSO 算法中的分数阶次 α 的动态调整方案中,提出一种带变异概率因子的群活性反馈的 S 型混合自适应分数阶粒子群优化方法 SFOPSO,并通过基准测试函数对算法的有效性和可行性进行了验证.

1 分数阶粒子群优化算法(FOPSO)

PSO 是模拟鸟类觅食行为的、基于群的随机优化算法,算法将鸟群中的个体看作一个没有重量和体积的粒子,算法目标是计算鸟群与食物的最短距离,在算法迭代过程中,算法中粒子(particle)根据当前群(swarm)中的全局最优及其个体自身历史最优位置更新其自身的位置和速度,经过若干次迭代后得到问题全局最优解. PSO 算法中群规模 M 、影响粒子速度变化程度的惯性权重 ω 、表示粒子对自身以及全局的依赖程度的两个学习因子 c_1, c_2 .

将分数阶微分的定义运用到 PSO 算法,改变粒子的速度和位置的更新方式,于是算法中每个粒子具有 FOC 分数阶微分的记忆特性,在 PSO 粒子的个体更新中融合其个体历史轨迹信息,从而使 FOPSO 算法具有更好的收敛性. α 阶 Grünwald-Letnikov 分数阶导数的表达式为

$$D^\alpha[x(t)] = \lim_{h \rightarrow 0} \left[\frac{1}{h^\alpha} \sum_{k=0}^{+\infty} \frac{(-1)^k \Gamma(\alpha+1) x(t-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \right] \quad (1)$$

在现实的离散时间条件下,式(1)可被定义为

$$D^\alpha[x(t)] = \frac{1}{T^\alpha} \left[\sum_{k=0}^r \frac{(-1)^k \Gamma(\alpha+1) x(t-kT)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \right] \quad (2)$$

式中, T 和 r 分别为采样周期和截断阶数. 根据 PSO 算法公式,如果令 $\omega=1, y_1=c_1 \times r_1, y_2=c_2 \times r_2$,则得到

$$v(t+1)_{i,j} + y_1(p_{\text{best}_{i,j}} - x(t)_{i,j}) + y_2(g_{\text{best}_j} - x(t)_{i,j}) \quad (3)$$

可以将其改写为

$$v(t+1)_{i,j} - v(t)_{i,j} = y_1(p_{\text{best}_{i,j}} - x(t)_{i,j}) + y_2(g_{\text{best}_j} - x(t)_{i,j}) \quad (4)$$

当采样周期 $T=1$ 时,式(4)的左边可写为

$$D^\alpha[x(t)] = [v(t+1)_{i,j}] = y_1(p_{\text{best}_{i,j}} - x(t)_{i,j}) + y_2(g_{\text{best}_j} - x(t)_{i,j}) \quad (5)$$

在 PSO 算法的迭代过程中,群中粒子与其前若干代的关系逐渐淡化,假如只保留前 4 代的速度信息可取截断阶数 $r=4$,由式(2)和(5)可以得到分数阶粒子群优化(FOPSO)的粒子个体速度和位置的更新公式为

$$v(t+1)_{i,j} = \alpha v(t)_{i,j} + \frac{1}{2} \alpha (1-\alpha) v(t-1)_{i,j} + \frac{1}{26} \alpha (1-\alpha) (2-\alpha) v(t-2)_{i,j} + \frac{1}{24} \alpha (1-\alpha) (2-\alpha) v(t-3)_{i,j} + y_1(p_{\text{best}_{i,j}} - x(t)_{i,j}) + y_2(g_{\text{best}_j} - x(t)_{i,j}) \quad (6)$$

由文献[12,17]的研究表明,一般地为了保证 FOPSO 算法具有更好的收敛性能,分数阶次 α 应当约束在 $[0.3, 0.8]$ 内取值,并利用下式调整 α

$$\alpha(t) = 0.8 - 0.5 \times \frac{t}{T} \quad (7)$$

式中, t 表示当前迭代次数、 T 表示总迭代次数.

2 群活性反馈的自适应分数阶粒子群优化

2.1 群活性(swarm activity)

在算法 FOPSO 中,粒子在 N 维空间内迭代运动,粒子的运动速度决定了粒子位置的移动幅度,同时也反映了 PSO 粒子在算法迭代过程中的活跃程度,而粒子的活跃程度的量化实际上体现着种群的多样性(diversity)以及种群的可探索性(exploitability)^[19-20],于是粒子个体和种群的活跃程度(activity)进行速度均方根表示为

$$\text{act}(t)_i = \sqrt{\frac{1}{N} \sum_{j=1}^N v^2(t)_j} \quad (8)$$

$$\text{Act}(t) = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N v^2(t)_{i,j}} \quad (9)$$

式中, $\text{act}(t)_i$ 表示算法中第 i 个粒子在第 t 代的活

性度,值越高表示 PSO 粒子越活跃、粒子位置离当前最优解越远,也表明该粒子的发掘能力越强。文献[18-19]利用 PSO 粒子的随机过程和移动中心吸引子的方法证明了粒子群优化的抽样分布及其群活性的有效性和稳定性。Act(t)表示算法中第 t 代时由 M 个粒子组成的种群 swarm 的群活性度,值越低表明 PSO 群中粒子多样性越差,此时种群中的粒子也相对集中。

2.2 分数阶次的动态调整 S 型策略

通过式(7)可知分数阶算法 FOPSO 中的分数阶次 α 在既定区间[0.3, 0.8]内呈线性地递减变化着,这种分数阶次的调整方法可致使 FOPSO 算法易出现早熟收敛现象,这种情况下 swarm 在 PSO 算法迭代前阶段由于粒子运动速度较快易导致整个 swarm 中的各粒子快速聚集于局部最优位置而出现“早熟 (premature convergence)”,而在算法迭代后阶段又由于 α 值又相对较小,削弱了粒子以及整个 swarm 逃脱局部最优的“自拔”能力。因此依据群活性值的变化情况将算法的 S 型状态分为探索期 (exploitation) 和开发期 (exploration),在 FOPSO 中引入函数 Sigmoid() 结合函数 Tanh() 实现 FOPSO 算法中的分数阶次 α 自适应地动态调整。

在算法探索期间,先用较低分数阶次值 α 约束群活性值相对较高 PSO 粒子的运动速度,以抑制粒子的快速聚集,从而增强 swarm 的全局搜索能力,然后将较高的 α 值用于活性值相对较低的 FOPSO 粒子以增强其逃脱局部最优的能力。如果其全局最优 (位置或速度) 在一定迭代期间的变化小于给定阈值甚至不再改变,则算法进入开发期,这期间将较高的 α 值用于群活性相对高的粒子以维持其高速度运动接近问题的全局最优位置,而将较小的分数阶次 α 用于活性度相对低的粒子以维持其以较低速度继续在探索邻域内最优位置,具体如下:

(I) 统计计算粒子活性度 $R(t)_i$ 为

$$R(t)_i = \frac{\text{act}(t)_i}{\text{Act}(t)} \quad (10)$$

(II) 当算法处于探索期时,分数阶次 α 的调整方案为

$$\alpha(t+1)_i = \frac{1}{1 + e^{-\frac{1}{R(t)_i}}} - \frac{1}{5}(1 - \tanh(R(t)_i)) \quad (11)$$

(III) 而当 FOPSO 在给定的多个迭代周期内 swarm 最优值几乎不再改变时,则算法运行开发期模式,分数阶次 α 在[0.3, 0.8]之间动态变化由式

$$\alpha(t+1)_i = \frac{1}{1 + e^{-\frac{1}{R(t)_i}}} - \frac{1}{5}(1 - \tanh(\frac{1}{R(t)_i})) \quad (12)$$

加以调整。

由式(11)与(12)可知,在 FOPSO 中采用上述策略后算法处在探索期与开发期两种交替状态下,swarm 中 α 在 Sigmoid() 与 Tanh() 的共同作用下动态变化,而且变化范围正处于[0.3, 0.8]之间,这与式(7)所述的分数阶约束相符合。

2.3 混合变异的自适应策略

虽然分数阶次 α 调整策略在 FOPSO 探索期间增强了种群的全局寻优能力、同时降低了算法早熟收敛的概率,而且由于算法在开采期内的 α 调整中着重增强了粒子的发掘能力,但这实际上在一定程度上又限制了 PSO 群中粒子逃脱局部最优的能力。即便如此,经过若干代迭代后分数阶 PSO 算法依然存在陷入早熟的可能性,为此我们引入一种混合变异操作算子让 FOPSO 算法迭代过程中 swarm 及其粒子也有机会自主逃脱当前局部最优,动态调整进化概率因子 β 再次减小算法早熟的可能,详细地步骤为:

Step1 设置进化概率因子的初值 $\beta=0$;

Step2 在 FOPSO 探索期,当种群最优位置连续不再变化时, $\beta=\beta+b_1$;

Step3 在 FOPSO 开采期里,当种群最优位置连续不再变化时, $\beta=\beta+b_2$;

Step4 由下式对当前 g_{best} 随机变异操作,同时重置 $\beta=0$;若 $\text{rand}()<\beta$,令

$$g_{\text{best}}(t)_j = g_{\text{best}}(t)_j \times \left(\frac{1}{2} + \eta \tanh(\text{Act}_y(t)) \right) \quad (13)$$

变异算子中 b_1, b_2 为[0.01, 0.1]之间的随机值以改变概率因子 β 。式(13)中的 η 是服从 Gauss(0, 1)分布的随机数,由于 $0 \leq \tanh(\text{Act}(t)) \leq 1$,那么这种变异策略使得全局最优值 g_{best} 的第 j 维在 $[0.5g_{\text{best}j}, 1.5g_{\text{best}j}]$ 内动态变化。

2.4 SFOPSO 算法

综上所述,我们提出群活性反馈的变异自适应分数阶粒子群优化算法 (SFOPSO),该算法利用 PSO 群活性度量值以动态地调整分数阶次 α ,在继承分数阶粒子群优化算法中粒子记忆优势的同时增强群中粒子在迭代时期的不同阶段全局探索能力与发掘能力,本文设计新的种群进化方案,简化了计算复杂度,详细步骤:

Step1 设置粒子群规模 M 、最大迭代次数 T 、搜索空间范围 $[X_{\min}, X_{\max}]$ 、学习因子 c_1, c_2 ,每个粒子位置 X_{ij} 与速度 V_{ij} 的初值、个体最优 p_{best} 和种群最优 g_{best} 可为当前最优位置;

Step2 由式(10)~(12)计算和调整分数阶次 α 值;

Step3 由式(6)和式(1)更新粒子速度和位置;

Step4 依据目标函数计算粒子适应度值并更新每个粒子的 p_{best} 和种群最优 g_{best} ;

Step5 判断当前算法状态,依据 2.3 节方法执行混合变异操作;

Step6 若符合给定终止条件或到达最大迭代次数,则算法终止运行、输出问题最优解,否则返回 Step2。

2.5 SFOPSO 算法收敛性分析

借鉴文献[15]的分析方法,若令式(6)中的 $p_{\text{best}i,j}$ 为 h_1 , $g_{\text{best}j}$ 为 h_2 ,则式(6)、(1)可改写成

$$v_{i+1} = ac_1 + \frac{1}{2}a(1-a)v_{i-1} +$$

$$\begin{aligned} & \frac{1}{6}a(1-a)(2-a)v_{t-2} + \\ & \frac{1}{24}a(1-a)(2-a)(3-a)v_{t-3} + \\ & y_1(h_1 - x_t) + y_2(h_2 - x_t) \end{aligned} \quad (14)$$

由 PSO 算法公式可知 $x_t = x_{t-1} + v_t$, 即

$$v_t = x_t - x_{t-1} \quad (15)$$

将式(15)代入式(14)可得

$$\begin{aligned} x_{t+1} = & x_t + a(x_t - x_{t-1}) + \\ & \frac{1}{2}a(1-a)(x_{t-1} - x_{t-2}) + \\ & \frac{1}{6}a(1-a)(2-a)(x_{t-2} - x_{t-3}) + \end{aligned}$$

$$\begin{aligned} & \frac{1}{24}a(1-a)(2-a)(3-a)(x_{t-3} - x_{t-4}) + \\ & y_1(h_1 - x_t) + y_2(h_2 - x_t) \end{aligned} \quad (17)$$

整理式(17), 可得

$$\begin{aligned} x_{t+1} = & (1 + a - y_1 - y_2)x_t + y_1h_1 + y_2h_2 - \\ & \frac{1}{2}ax_{t-1} - \frac{1}{3}ax_{t-2} - \\ & \left[\frac{1}{6}a - \frac{1}{24}a(1-a)(2-a) \right] x_{t-3} - \\ & \frac{1}{24}a(1-a)(2-a)x_{t-4} \end{aligned} \quad (18)$$

将该表达式写成矩阵向量乘积形式

$$\begin{bmatrix} x_{t+1} \\ x_t \\ x_{t-1} \\ x_{t-2} \\ x_{t-3} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + a - y_1 - y_2 & \frac{1}{2}a & \frac{1}{3}a & - \left[\frac{1}{6}a - \frac{1}{24}a(1-a)(2-a) \right] & \frac{1}{24}a(1-a)(2-a) & y_1z_1 + y_2z_2 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \\ x_{t-3} \\ x_{t-4} \\ 1 \end{bmatrix}$$

求得该矩阵的特征多项式为

$$\begin{aligned} & [\lambda^4(1 + a - y_1 - y_2) - \frac{1}{2}a\lambda(\lambda^2 + 1) + \\ & \frac{1}{24}a(1-a)(2-a)(1 + \lambda)](1 - \lambda) \end{aligned} \quad (19)$$

显然, $\lambda_1 = 1.0$ 为式(19)的一个解, 若另外 4 个解分别表示为 $\lambda_i, i = 2 \sim 5$, 则在 t 时刻的式(15)可写成

$$x_t = c_1 + c_2\lambda_2 + c_3\lambda_3 + c_4\lambda_4 + c_5\lambda_5 \quad (20)$$

式中, 常数 $c_i (i = 1 \sim 5)$, 为了证明粒子位置过程 $\{x_t\}$ 的收敛性, 即式(20)的极限形式

$$\lim_{t \rightarrow \infty} x_t = \lim_{t \rightarrow \infty} (c_1 + c_2\lambda_2 + c_3\lambda_3 + c_4\lambda_4 + c_5\lambda_5) \quad (21)$$

由此可得 $\{x_t\}$ 收敛条件是式(21)收敛于 c_1 , 即

$$\lim_{t \rightarrow \infty} x_t = c_1,$$

也即

$$\lim_{t \rightarrow \infty} (c_2\lambda_2 + c_3\lambda_3 + c_4\lambda_4 + c_5\lambda_5) = 0 \quad (22)$$

要使得式(22)成立, 则必须有 $\max(\|\lambda_i\|) < 1$, 其中 $i = 2 \sim 5$, 所以必须证明如果 $\|\lambda_i\| > 1$, 式(19)均不成立。

将 $\|\lambda_2\| > 1$ 代入特征多项式(19)得

$$\begin{aligned} & [\lambda_2^4(1 + a - y_1 - y_2) - \frac{1}{2}a\lambda_2(\lambda_2^2 + 1) + \\ & \frac{1}{24}a(1-a)(2-a)(1 + \lambda_2)](1 - \lambda_2) \end{aligned} \quad (23)$$

式中, $y_1, y_2 \in [0, 1.5], a(1-a)(2-a)$ 在 0.3 到 0.8 的闭区间上单调递减的, 其最大函数值为 0.357, 且函数式 $(1 + a - y_1 - y_2)$ 的值域为 $[-1.7, 1.8]$, 那么可得函数 $f(\lambda_2)$

$$f(\lambda_2) = [\lambda_2^4(1 + a - y_1 - y_2) - \frac{1}{2}a\lambda_2(\lambda_2^2 + 1) +$$

$$\frac{1}{24}a(1-a)(2-a)(1 + \lambda_2)](1 - \lambda_2) \quad (24)$$

由此求解可得 $-5 + 5\lambda_2 \geq f(\lambda_2) \geq -0.25 + 0.25\lambda_2 > 0$, 这与 $\|\lambda_2\| > 1$ 时 $f(\lambda_2) = 0$ 显然是矛盾的, 所以假设不成立, 也就是 $\lambda_i (i = 2, 3, 4, 5)$ 时 $\|\lambda_i\| < 1$, 因此在当前参数设置的条件下本文提出的 SFOPSO 算法是收敛的。

3 实验结果分析

3.1 实验设计

实验中选取 6 个 Benchmarks 函数如表 1 所示, 6 个函数是具有不同峰值特征的单目标基准测试函数, 并选用 3 个算法 DPSO、FOPSO、FODPSO 进行对比实验, 同时还将使用未采取混合变异自适应策略的 SFOPSO 算法 (SFOPSO-N) 作为对照算法进行比较, 以验证本文算法采用混合变异的自适应策略 SFOPSO 算法的综合寻优性能。实验环境为 Windows 10 下 Python 3.5.1 编程实现的^[6]。

3.2 实验测试结果分析

各算法参数设置如表 2 所示, 所有算法在每个函数上均重复运行 30 次记录并统计运行结果如表 3 所示。相比 3 种方法 DPSO、FOPSO 以及 FODPSO, 本文提出的 SFOPSO 算法具有更好的收敛精度以及算法稳定性, 尤其是在多峰函数上 SDOPSO 的性能优势更为明显。由表 3 数据结合图 1~6, 实验中的算法 DPSO 和 FOPSO 对有些函数如特别是在多峰 Rosenbrock 等的寻优过程中, 在并未到达最大迭代次数前已处于停滞状态, 这时全局最优就不再继续改变, 而对于这些函数算法 FODPSO 和 SFOPSO 即使在最大迭代次数之后, 仍然保持着向实际最优位置逐渐逼近。同时, 通过对比 SFOPSO 算法与 SFOPSO-N 算法的实验结果可知, 本文设计的混合变异自适应策略可以有效

地促进种群自主逃离当前最优位置,降低早熟收敛 的风险,提升算法的收敛精度.

表 1 测试函数

Tab. 1 Test functions

Functions	Formulations	Dimensions	Domain	Optional
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-50, 50]$	$f_1^*(0) = 0$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1))$	30	$[-10, 10]$	$f_2^*(0) = 0$
DeJong F4	$f_3(x) = \sum_{i=1}^n x_i^4$	30	$[-20, 20]$	$f_3^*(0) = 0$
Ackley	$f_4(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]$	$f_4^*(0) = 0$
Griewank	$f_5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]$	$f_5^*(0) = 0$
Rastrigin	$f_6(x) = \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) + 10$	30	$[5.12, 5.12]$	$f_6^*(0) = 0$

表 2 实验算法参数设置

Tab. 2 Parameters setting for experimental algorithms

Parameters	DPSO/FOPSO /FODPSO	SFOPSO/SFOPSO-N
Swarm size, M	30	30
Maximum iteration, T	1000	1000
Learning factors c_1, c_2	1.49, 1.49	1.49, 1.49
Fractional order, α	Decrease linearly by iterations in $[0.3, 0.8]$	Change nonlinearly in $[0.3, 0.8]$ by Eqs. (6)~(8)
Sub-swarm sizes	5 sub-swarms, $4 \leq$ each sub-swarm size ≤ 8	

表 3 实验计算结果

Tab. 3 Experimental results

Functions		Algorithms				
		DPSO	FOPSO	FODPSO	SFOPSO-N	SFOPSO
Sphere	Mean	3.110e-21	1.643e-20	5.997e-21	2.884e-19	1.559e-21
	SD	4.24e-11	4.444e-15	2.542e-19	6.232e-16	5.076e-25
Ackley	Mean	3.0797	1.2079	1.476e-7	0.1438	9.698e-12
	SD	6.4558	2.9197	0.559e-5	1.2886	4.641e-11
DeJong	Mean	1.587e-10	9.253e-39	3.786e-39	8.475e-38	8.969e-47
	SD	3.455e-7	4.346e-37	5.453e-38	9.471e-36	9.4947e-46
Rosenbrock	Mean	52.2229	22.2231	20.8119	19.2138	15.7907
	SD	70.5308	45.2337	40.4556	43.1262	38.9209
Rastrigin	Mean	5.2329	3.322e-4	6.347e-6	1.239e-5	1.768e-13
	SD	8.2339	0.1519	4.261e-3	9.876e-3	1.465e-11
Griewank	Mean	4.821e-4	2.123e-8	8.756e-11	1.893e-7	9.756e-14
	SD	0.9237	2.783e-4	4.562e-8	3.235e-4	8.754e-14

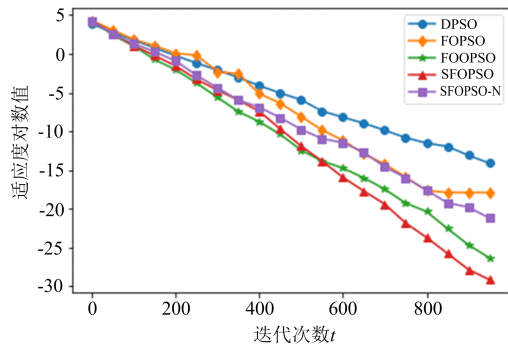


图 1 在 Sphere 上的收敛曲线
Fig. 1 Fitness changes on Sphere

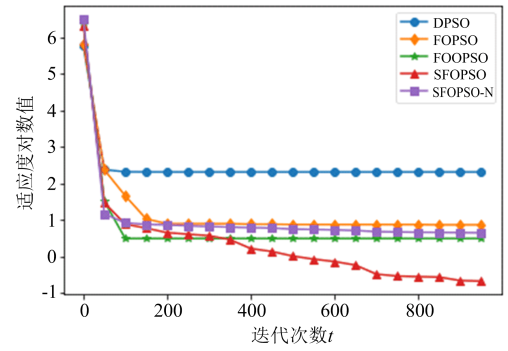


图 2 在 Rosenbrock 上的收敛曲线
Fig. 2 Fitness changes on Rosenbrock

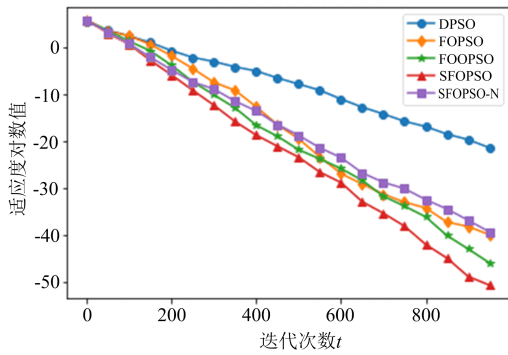


图 3 在 DeJong F4 上的收敛曲线
Fig. 3 Fitness changes on DeJong F4

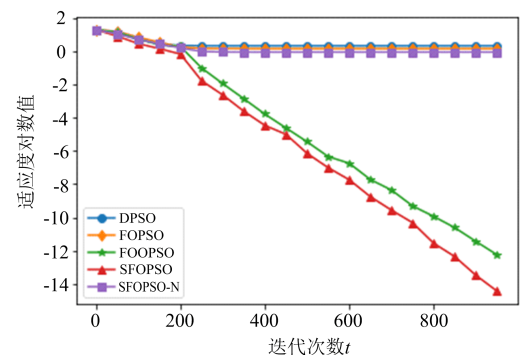


图 4 在 Ackley 上的收敛曲线
Fig. 4 Fitness changes on Ackley

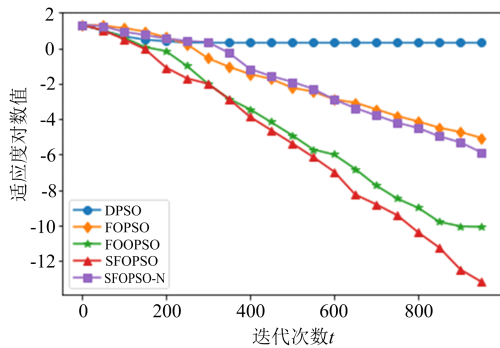


图 5 在 Griewank 上的收敛曲线
Fig. 5 Fitness changes on Griewank

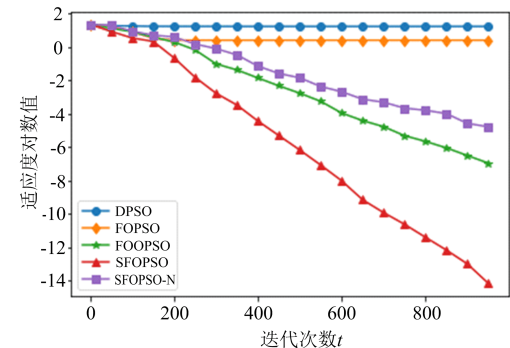


图 6 在 Rastrigin 上的收敛曲线
Fig. 6 Fitness changes on Rastrigin

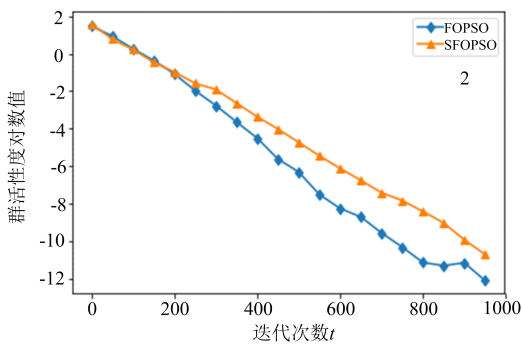


图 7 在 Sphere 上的群活性值变化
Fig. 7 Swarm activities on Sphere

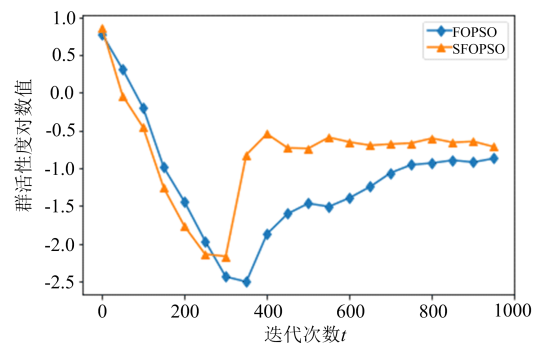


图 8 在 Rosenbrock 上的群活性值变化
Fig. 8 Swarm activities on Rosenbrock

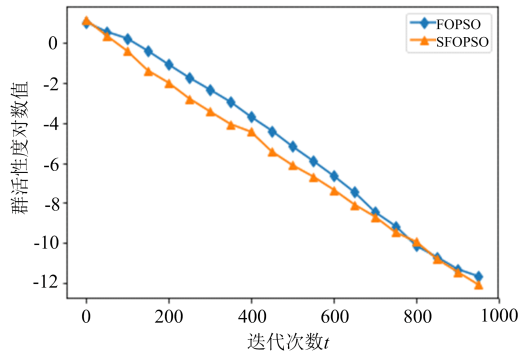


图 9 在 DeJong F_4 上的群活性值变化
Fig. 9 Swarm activities on DeJong F_4

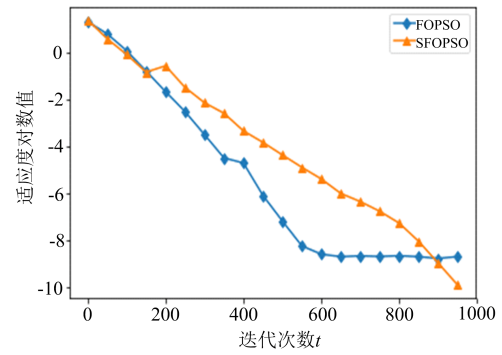


图 10 在 Ackley 上的群活性值变化
Fig. 10 Swarm activities on Ackley

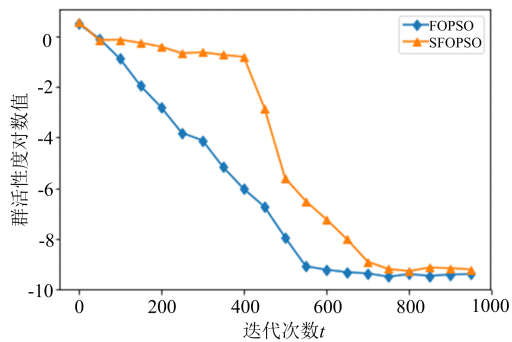


图 11 在 Griewank 上的群活性值变化
Fig. 11 Swarm activities on Griewank

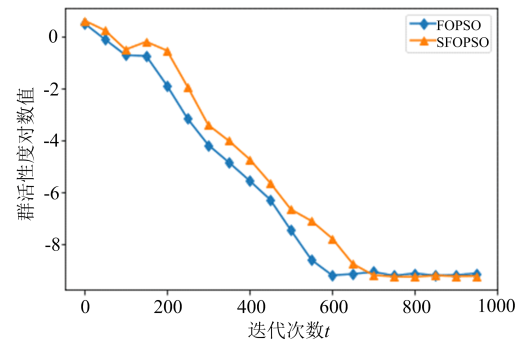


图 12 在 Rastrigin 上的群活性值变化
Fig. 12 Swarm activities on Rastrigin

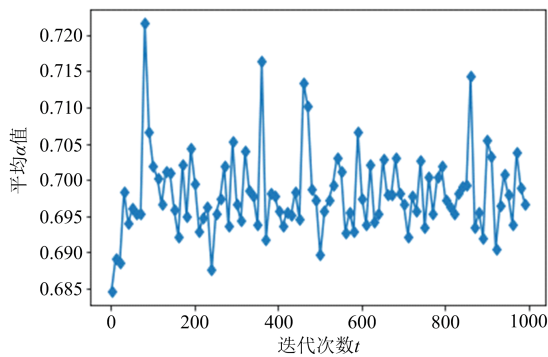


图 13 在 Sphere 上平均 α 值变化
Fig. 13 Means of fractional order α for Sphere

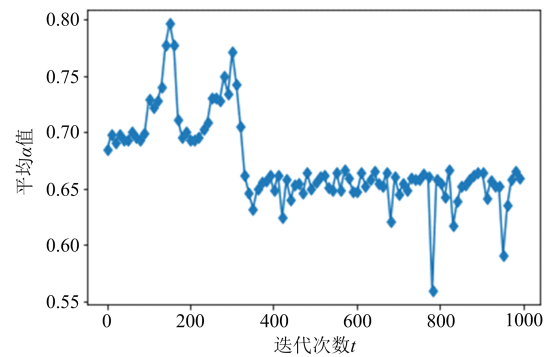


图 14 在 Rosenbrock 上平均 α 值变化
Fig. 14 Means of fractional order α for Rosenbrock

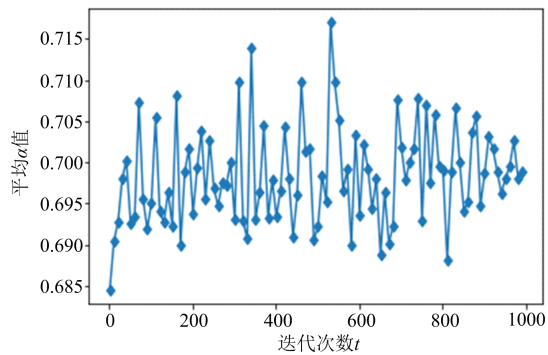


图 15 在 DeJong F_4 上平均 α 值变化
Fig. 15 Means of fractional order α for DeJong F_4

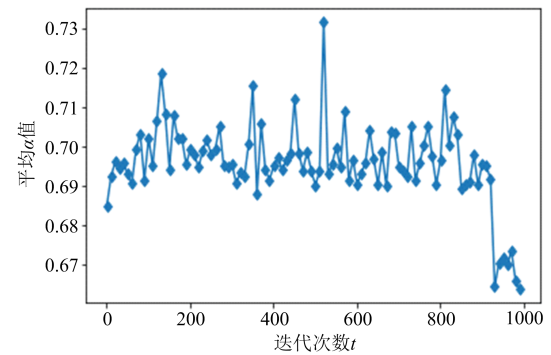


图 16 在 Ackley 上平均 α 值变化
Fig. 16 Means of fractional order α for Ackley

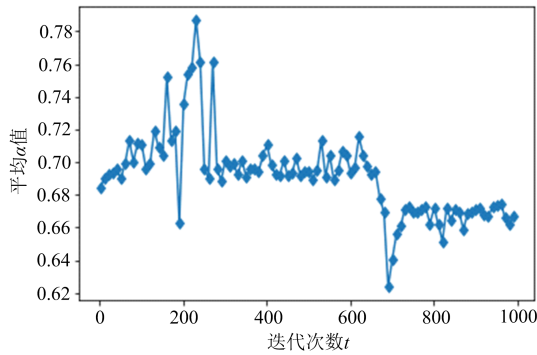
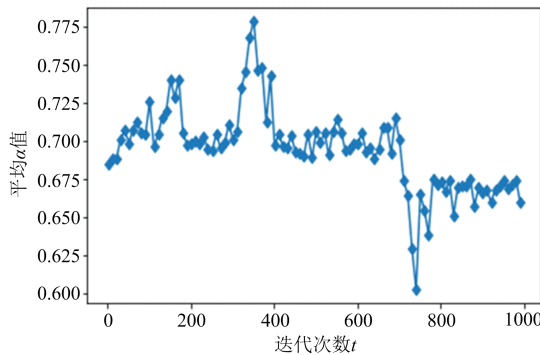
图 17 在 Griewank 上平均 α 值变化Fig. 17 Means of fractional order α for Griewank图 18 在 Rastrigin 上平均 α 值变化Fig. 18 Means of fractional order α for Rastrigin

图 1~6 显示了 5 种算法在 Sphere 等 6 个基准测试函数上的收敛过程,由图可知,SFOPSO 在算法运行寻优搜索中能够较快地收敛.图 7~12 分别展示了算法 FOPSO 和 SFOPSO 在 Griewank 等 6 个函数上群活性量值的变化态势,群中粒子稳定地向问题的实际全局最优位置逐步接近,表明 SFOPSO 在迭代过程中粒子群活性整体呈现平稳递减的趋势;当粒子群逃脱局部最优位置时,群活性值的变化程度也发生跳跃变化,而算法 FOPSO 的群活性度则迅速下降,随迭代次数的增加线性递减,实际上在对 Ackley 函数求解时,群活性度值在 600 代以后已经趋于稳定,这反映了 SFOPSO 算法在迭代运行中保持了恰当的群活性和粒子多样性,因而算法在搜索后期仍然能维持着较强的寻优搜索能力.图 13~18 显示了 SFOPSO 算法中 S 型分数阶调整策略在对 Ackley 等 6 个函数求解时 α 的动态,在算法迭代过程中粒子的 α 平均值始终在 $[0.65, 0.75]$ 之间振荡,表明依据种群活性以及粒子的活性值的变化自适应动态调整粒子的分数阶次是可行的,在算法运行过程中保持适当的群活性和粒子多样性,能有效地增强种群逃脱局部最优的能力.

4 结论

在分析分数阶粒子群算法原理的基础上,针对传统分数阶粒子群(FOPSO)算法在算法综合性能上依赖于分数阶次 α ,易陷入早熟收敛的问题,本文提出了一种基于群活性反馈的 S 型自适应分数阶

粒子群方法(SFOPSO),根据种群活性以及粒子个体的活跃程度自适应地动态调整每个粒子的分数阶次 α ,使种群在搜索过程中保持较好的稳定性与多样性;同时设计了一种混合变异机制增强粒子种群跳出局部最优的能力、进一步降低了算法陷入早熟收敛的可能.理论分析证明了本文提出的算法 SFOPSO 的收敛性,6 个不同峰值特征的基准优化函数实验测试结果表明,本文提出的 SFOPSO 算法是可行和有效的,在未增加计算复杂度的同时能够使粒子群在算法迭代过程的不同阶段保持较强的全局搜索能力和在搜索后阶段保持较强的邻域开采能力.最后将本文提出的算法与基于达尔文进化思想 FODPSO 等 4 种方法相比较,结果表明, SFOPSO 具有更优的收敛精度和更快的收敛速度.未来将进一步挖掘分数阶记忆历史轨迹信息的特征优势,针对工程应用实际问题开发出可行、高效、高精度的群智能优化方法.

参考文献(References)

- [1] JAIN N K, NANGIA U, JAIN J. A review of particle swarm optimization [J]. Journal of The Institution of Engineers (India): Series B, 2018, 99(4):407-411.
- [2] HARRISON K R, OMBUKI-BERMAN B M, ENGELLBECHT A P. A parameter-free particle swarm optimization algorithm using performance classifiers [J]. Information Sciences, 2019, 503(7): 381-400.
- [3] SU S B, GUO H F, TIAN H M, et al. A novel pattern clustering algorithm based on particle swarm optimization joint adaptive wavelet neural network model [J]. Mobile Networks and Applications, 2017, 22(4): 692-701.
- [4] ISIET M, GADALA M. Self-adapting control parameters in particle swarm optimization[J]. Applied Soft Computing Journal, 2019, 83(10):105653.
- [5] GHAMISI P, COUCEIRO M S, BENEDIKTSSON J A, et al. An efficient method for segmentation of images based on fractional calculus and natural selection[J]. Expert Systems with Applications, 2012, 39(16):12407-12417.
- [6] 陈秋鑫. 分数阶粒子群优化及其在聚类分析中的应用 [D]. 镇江:江苏科技大学, 2018.
CHEN Q X. Fractional-order particle swarm optimization and its applications in clustering analysis [D]. Zhenjiang: Jiangsu University of Science and Technology, 2018.
- [7] 翟兆睿, 苏守宝. 一种动态压缩因子的分数阶粒子群优化 [J]. 重庆理工大学学报(自然科学), 2019, 33(7): 94-101.
ZHAI Z R, SU S B. A fractional-order particle swarm optimization with dynamic constriction factor [J]. Journal of Chongqing University of Technology (Natural Science), 2019, 33(7): 94-101.
- [8] BENSON D A, MEERSCHAERT M M, REVIELLE J. Fractional calculus in hydrologic modeling: A numerical perspective [J]. Advances in Water Resources, 2013, 51(1): 479-497.
- [9] ZHANG Y, SUN H G, STOWELL H H, et al. A review of applications of fractional calculus in Earth

- system dynamics [J]. *Chaos Solitons & Fractals*, 2017, 102: 29-46.
- [10] PIRES E J S, MACHADO J A T, OLIVEIRA P B M, et al. Fractional order dynamics in a particle swarm optimization algorithm[C]// *International Conference on Intelligent Systems Design and Applications*. IEEE, 2007: 703-710.
- [11] PIRES E J S, MACHADO J A T, OLIVEIRA P B M, et al. Particle swarm optimization with fractional-order velocity[J]. *Nonlinear Dynamics*, 2010, 61(1-2): 295-301.
- [12] COUCEIRO M S, GHAMISI P. *Fractional Order Darwinian Particle Swarm Optimization* [M]. Springer, 2016.
- [13] COUCEIRO M S, GHAMISI P. Fractional order darwinian particle swarm optimization [J]. *Applied Sciences & Technology*, 2015, 2011(11): 127-136.
- [14] COUCEIRO M S. Introducing the fractional-order darwinian PSO[J]. *Signal Image & Video Processing*, 2012, 6(3): 343-350.
- [15] ZHANG X, DUAN H. Comments on Particle swarm optimization with fractional-order velocity [J]. *Nonlinear Dynamics*, 2014, 77(1-2): 427-429.
- [16] 郭通, 兰巨龙, 李玉峰, 等. 自适应的分数阶达尔文粒子群优化算法[J]. *通信学报*, 2014, 35(4): 130-140.
GUO T, LAN J L, LI Y F, et al. Adaptive fractional-order darwinian particle swarm optimization algorithm [J]. *Journal of Communications*, 2014, 35(4): 130-140.
- [17] YU S, SU S B, HUANG L. A simple diversity guided firefly algorithm [J]. *Kybernetes*, 2015, 44(1): 43-56.
- [18] SU S B, CHEN Q X, ZHAI Z R. Fractional-order particle swarm optimization with swarm activity feedback [J]. *Basic & Clinical Pharmacology & Toxicology*, 2019, 124(S2): 98-99.
- [19] 苏守宝, 曹喜滨, 孔敏. 群活性与粒子群优化的稳定性分析[J]. *控制理论与应用*, 2010, 27(10): 1411-1417.
SU S B, CAO X B, KONG M. Stability analysis of particle swarm optimization using swarm activity [J]. *Control Theory and Applications*, 2010, 27(10): 1411-1417.
- [20] Huang L, DU W W, Ding L X. Particle swarm optimization algorithm based on adaptive Sigmoid inertia weight[J]. *Application Research of Computers*, 2012, 29(1): 32-34.