

doi: 10.3969/j.issn.0490-6756.2018.05.008

基于分数阶位置状态的量子粒子群算法

徐 来, 蒲亦非, 周激流

(四川大学计算机学院, 成都 610065)

摘要: 由量子力学的概念和粒子群优化算法的结合, 量子行为粒子优化算法作为粒子群算法的一个变种, 具有更好的全局搜索能力. 为了提高量子粒子算法的全局搜索能力, 结合分数阶微积分的概念, 本文提出了一种新的算法. 该算法将分数阶微积分中常用的 GL 定义引入了量子粒子算法的更新迭代公式中, 利用分数阶微积分的长时记忆特性, 通过记忆量子粒子在更新迭代过程中的历史位置和历史信息, 增强算法的收敛速度和收敛精度. 为了全面评估算法的基本性能, 本文进行了一些关于基本测试函数的功能测试. 通过对于不同阶次的分数阶量子粒子算法的对比实验和与其他粒子群改进算法的对比实验, 实验结果表明, 该算法具有更高的收敛精度.

关键词: 量子粒子群; 分数阶; 长时记忆性; GL 定义

中图分类号: TP309.7 **文献标识码:** A **文章编号:** 0490-6756(2018)05-0947-08

An improved quantum particle swarm optimization with fractional-order position

XU Lai, PU Yi-Fei, ZHOU Ji-Liu

(College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: Motivated by concepts in quantum mechanics and particle swarm optimization (PSO), quantum-behaved particle swarm optimization was proposed as a variant of PSO with better global search capability. This paper proposes a novel method for enhancing the global search capability of PSO and guiding its search with fractional calculus concepts. With the commonly used definitions of fractional differential known as Grünwald-Letnikov (GL), the authors introduce its discrete expression into the position update in QPSO to improve its convergence speed and accuracy. Some empirical studies on popular benchmark functions are performed in order to make a full evaluation on performance and comparison between standard QPSO and QPSO with different fractional-order. The new algorithm, named fractional-order Quantum particle swarm optimization, shows to perform well in finding optimal solutions with much higher convergence accuracy in many optimization problems.

Keywords: Quantum particle swarm optimization; Fractional-order; Long-time memory; GL definition

1 引言

基于鸟群和鱼群的社会行为理论的粒子群优

化算法^[1]首先由 Kennedy 和 Eberhart 作为群智能技术被提出. 在 PSO 算法中, 潜在的解决方案称为粒子, 通过跟随自己的经验和当前最好的粒子, 飞

收稿日期: 2018-06-01

基金项目: 国家自然科学基金(61571312)

作者简介: 徐来(1994-), 男, 四川内江人, 硕士生, 研究方向为智能信息处理. E-mail: 594107185@qq.com;

通讯作者: 蒲亦非. E-mail: puyifei_007@163.com

过问题空间. PSO 算法在性能上与众所周知的遗传算法(GA, Genetic Algorithm)^[2]相当. 其在执行复杂的优化任务中的有效性, 在过去 10 年中已经越来越受欢迎.

量子机械计算机^[3]在 20 世纪 80 年代被提出, 量子力学计算机的描述在 1980 年末正式化. 量子计算机的许多努力自 20 世纪 90 年代初以来一直在积极进行.

为了提高 PSO 的收敛性, Sun 等在 2004 年开发的基于标准 PSO 的量子行为粒子群优化算法(QPSO, Quantum Particle Swarm Optimization)^[4]. 基于量子空间, 粒子可以在整个可行解的空间中搜索, 从而可以获得全局最优解. 因此, QPSO 算法是保证全局收敛的算法, 在搜索能力方面优于原始 PSO 算法. 但与其它进化算法一样, QPSO 算法依然不可避免地会遇到早熟收敛的问题.

在过去的 300 多年中, 分数阶微积分^[5]已经越来越受到人们的关注, 成为了重要的数学分支. 然而, 它在工程领域却很少受到数学家和物理科学家所了解. 一般来说, 欧几里得空间的差分逐步将步长扩展到分数. 分数阶微积分是古典数学的一个重要分支. 其实从整数阶微积分理论建立开始之初, 几位数学家就研究了非整数阶导数和积分的计算. 不过到现在, 分数阶微积分的应用一直很少, 但是近几十年该领域受到了广泛的关注^[6,7].

虽然现在对 QPSO 有一些改进, 但是它们在遍历过程中不能充分利用信息, 所以在全局优化方面效果不是最佳. 本文提出了一种基于分数阶位置的量子粒子优化算法. 正如我们所知, 分数阶微分算子具有长时记忆性, 这使得 QPSO 更有助于获得全局优化.

本文将分数阶微积分引入了量子粒子群算法, 提出了一种改进的量子粒子群算法: 基于分数阶位置的量子粒子群算法. 利用分数阶微积分的长时记忆特性, 在量子空间中记录粒子的个体最优位置和群体最优位置. 本文的其余组织部分内容如下: 第 2 节介绍了关于分数演算的一些数学背景; 第 3 节介绍了基本的 PSO 和量子 PSO, 并且给出了所提出的方法; 第 4 节介绍了具有分数位置的 QPSO 的实验结果, 并分析了该算法的收敛速度和收敛精度; 最后, 第 5 节概述了主要结论.

2 数学背景

在欧式测度下常用的分数阶微积分的定义^[8]

有 3 种: GL 定义^[9], RL 定义和 Captuo 定义. 当然, 还有其他比较出名的分数阶定义.

分数阶微积分的 Grünwald-Letnikov 定义如下.

$${}^a D_x^\alpha f(x) = \frac{d^\alpha}{[d(x-a)]^\alpha} f(x) =$$

$$\lim_{n \rightarrow \infty} \left\{ \frac{(x-a/n)^{-\alpha}}{\Gamma(-\alpha)} \sum_{k=0}^{n-1} \frac{\Gamma(k-a)}{\Gamma(k+1)} f\left(x-k\left(\frac{x-a}{n}\right)\right) \right\} \quad (1)$$

从上面的定义可以看出分数阶微分与之前的所有历史状态有关, 具有长期连续记忆的特性.

离散形式的 Grünwald-Letnikov 定义如下.

$$\frac{d^\alpha}{dx^\alpha} f(x) \cong \frac{x^{-\alpha} N^\alpha}{\Gamma(x-a)} \sum_{k=0}^{N-1} \frac{\Gamma(k-a)}{\Gamma(k+1)} f\left(x - \frac{kx}{N}\right) \quad (2)$$

这种近似的表达式使得分数阶微分得以通过简单的乘除和加减实现. 对于一维信号, 其还有如下的表达方式:

$$\begin{aligned} \frac{d^\alpha}{dx^\alpha} f(x) \cong & f(x) + (-\alpha)f(x-1) + \\ & \frac{(-\alpha)(-\alpha+1)}{2} f(x-2) + \dots \\ & + \frac{\Gamma(n-\alpha-1)}{(n-1)! \Gamma(-\alpha)} f(x-n+1) \end{aligned} \quad (3)$$

3 分数阶量子粒子群

3.1 量子粒子群算法

有很多实验已经证明了 PSO 算法不能收敛于全局最优解^[10], 甚至于局部最优解, 许多学者已经采用了很多方法来改进粒子群优化算法的性能. 2004 年 Sun 等在研究了 Clerc 等人关于粒子收敛行为的研究成果后, 从量子力学的角度提出了一种新的 PSO 算法模型. 认为粒子具有量子行为, 并根据这种模型提出了量子粒子群算法(QPSO, Quantum Particle Swarm Optimization), 在量子的空间中粒子的满足聚集形态的性质完全不同, 它可以在整个可行解的空间中进行搜索, 因此 QPSO 算法的全局搜索性能远远优于标准 PSO 算法. 通过波函数来描述粒子的状态, 并通过求解薛定谔方程得到粒子在空间中的某一点的概率密度函数, 又通过蒙特卡罗随机模拟方式得到粒子的位置方程.

在量子粒子群中, 引入势降, 假设粒子是在以 p 点为中心的势降中, 因为粒子的速度和位置在量

子的空间中不能一起确定, 所以粒子的状态用波函数来表示.

$$\varphi(Y) = \frac{1}{\sqrt{L}} e^{-\frac{|Y|}{L}}, L = \frac{1}{\beta} = \frac{h^2}{mr} \quad (4)$$

其中, L 为粒子出现在相对的点位置的概率. 采用蒙特卡洛方法, 得出的粒子的位置方程为

$$X_{t+1} = p \pm \frac{L}{2} \ln\left(\frac{1}{u}\right) \quad (5)$$

其中, $u \sim u(0, 1)$; L 为势降的特征长度.

以上都是在一维势降下研究的粒子位置, 对于 n 维空间也类似. 设吸引子为: $P_i = (P_{i1}, P_{i2}, \dots, P_{in})$ 每一维以 P_{in} 为坐标, P_{ij} 为中心建立一维势阱, 对于 P_{ij} 有粒子 i 的每一维波函数:

$$\varphi[x_{ij}(t+1)] = \frac{1}{\sqrt{L_{ij}(t)}} \exp\left[-\frac{|x_{ij}(t+1) - P_{ij}(t)|}{L_{ij}(t)}\right] \quad (6)$$

位置方程为

$$x_{ij}(t+1) = P_{ij}(t) + \frac{L_{ij}(t)}{2} \ln\left[\frac{1}{u_{ij}(t)}\right] \quad (7a)$$

$$x_{ij}(t+1) = P_{ij}(t) - \frac{L_{ij}(t)}{2} \ln\left[\frac{1}{u_{ij}(t)}\right] \quad (7b)$$

$$u_{ij}(t) \sim u(0, 1)$$

引入平均最好位置(mean best position), 定义如下.

$$mbest = \frac{1}{M} \sum_{i=1}^n P_i(t) = \left(\frac{1}{M} \sum_{i=1}^M P_{i1}(t), \frac{1}{M} \sum_{i=1}^M P_{i2}(t), \dots, \frac{1}{M} \sum_{i=1}^M P_{im}(t)\right) \quad (8)$$

则有

$$L_{ij}(t) = 2\alpha |mbest_{j(c)} - x_j(t)| \quad (9)$$

则位置方程的更新公式为

$$x_{ij}(t+1) = P_{ij}(t) \pm \alpha |mbest_{j(c)} - x_j(t)| \times \ln(1/u_{ij}(t)) \quad (10)$$

$$u_{ij}(t) \sim u(0, 1)$$

其中, α 为收缩扩张因子, 把应用于以上公式的粒子群算法称为量子粒子群算法.

3.2 分数阶量子粒子群算法

在本节中, 我们将分数阶的长时记忆性^[11] 加入到了量子粒子群算法中, 得到了一种新的量子算法, 称之为分数阶量子粒子群算法(FQPSO, Fractional quantum particle swarm optimization). 首先, 根据量子粒子群算法的性质, 我们可以将量子迭代更新的公式改写成如下的形式:

$$x_{ij}(t+1) = P_{ij}(t) + \alpha * \ln(1/u) * (x_{ij}(t) - mbest_j(t)) \quad (11a)$$

$$(rand > 0.5, mbest_j < x_{ij}(t))$$

$$x_{ij}(t+1) = P_{ij}(t) - \alpha * \ln(1/u) * (mbest_j - x_{ij}(t)) \quad (11b)$$

$$(rand < 0.5, mbest_j > x_{ij}(t))$$

$$x_{ij}(t+1) = P_{ij}(t) + \alpha * \ln(1/u) * (mbest_j - x_{ij}(t)) \quad (11c)$$

$$(rand > 0.5, mbest_j > x_{ij}(t))$$

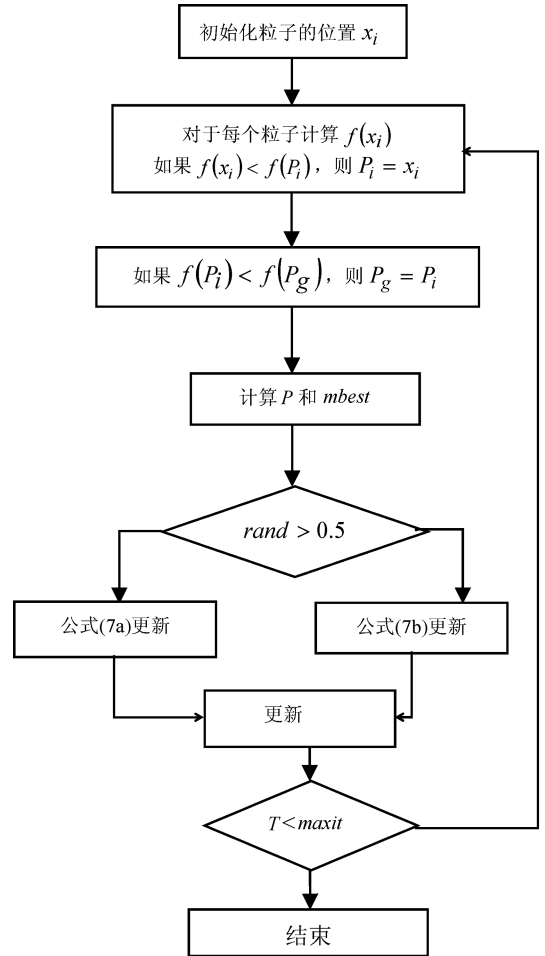


图 1 QPSO 算法流程图
Fig. 1 The flow chart of QPSO

$$x_{ij}(t+1) = P_{ij}(t) - \alpha * \ln(1/u) * (mbest_j - x_{ij}(t)) \quad (11d)$$

$$(rand < 0.5, mbest_j < x_{ij}(t))$$

公式也可以改写成以下的形式:

$$x_{ij}(t+1) - x_{ij}(t) = P_{ij}(t) + \alpha * \ln(1/u) * mbest_j - (\alpha * \ln(1/u) + 1) * x_{ij}(t) \quad (12a)$$

以上公式满足当 $rand > 0.5, mbest_j < x_{ij}(t)$ 或者 $rand < 0.5, mbest_j > x_{ij}(t)$ 两种情况时成立.

$$x_{ij}(t+1) - x_{ij}(t) = P_{ij}(t) - \alpha * \ln(1/u) * mbest_j + (\alpha * \ln(1/u) - 1) * x_{ij}(t) \quad (12b)$$

以上公式满足当 $rand > 0.5, mbest_j < x_{ij}(t)$ 或者 $rand < 0.5, mbest_j > x_{ij}(t)$ 两种情况时成立.

而等式左边可以理解成一个一阶差分算子,因此也可以改写成如下的表达形式:

$$D^{\alpha}(x_{ij}(t+1)) = P_{ij}(t) + \alpha * \ln(1/u) * mbest_j - (\alpha * \ln(1/u) + 1) * x_{ij}(t) \quad (13)$$

以上公式满足当 $rand > 0.5, mbest_j < x_{ij}(t)$ 或者 $rand < 0.5, mbest_j > x_{ij}(t)$ 两种情况时成立.

$$D^{\alpha}(x_{ij}(t+1)) = P_{ij}(t) - \alpha * \ln(1/u) * mbest_j + (\alpha * \ln(1/u) - 1) * x_{ij}(t) \quad (14)$$

以上公式满足当 $rand > 0.5, mbest_j < x_{ij}(t)$ 或者 $rand < 0.5, mbest_j > x_{ij}(t)$ 两种情况时成立.

当 α 不等于 1 时,量子粒子群算法迭代方程中的差分算子就可以由一阶推广到任意阶次^[11,12]. 这样,就使得量子粒子群算法具有了分数阶微分的长时记忆性.我们将公式(3)中令 $n=4$,则可得:

$$D^{\alpha}(x_{ij}(t+1)) = x_{ij}(t+1) - \alpha x_{ij}(t) - \frac{1}{2}\alpha(1-\alpha)x_{ij}(t-1) - \frac{1}{6}\alpha(1-\alpha)(2-\alpha)x_{ij}(t-2) - \frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)x_{ij}(t-3) \quad (15)$$

将其代入量子粒子群算法的更新迭代公式可以得到新的粒子更新迭代公式^[13-16],即基于分数阶位置的量子粒子群算法的粒子更新迭代公式:

$$x_{ij}(t+1) = P_{ij}(t) + \alpha * \ln\left(\frac{1}{u}\right) * mbest_j - (\alpha * \ln\left(\frac{1}{u}\right) + 1 - \alpha) x_{ij}(t) + \alpha x_{ij}(t) \quad (16a)$$

$$x_{ij}(t+1) = P_{ij}(t) - \alpha * \ln\left(\frac{1}{u}\right) * mbest_j + (\alpha * \ln\left(\frac{1}{u}\right) + 1 + \alpha) x_{ij}(t) + \alpha x_{ij}(t) \quad (16b)$$

其中,

$$\alpha x_{ij}(t) = \frac{1}{2}\alpha(1-\alpha)x_{ij}(t-1) + \frac{1}{6}\alpha(1-\alpha)(2-\alpha)x_{ij}(t-2) +$$

$$\frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)x_{ij}(t-3) \quad (17)$$

从式(16a)和式(16b)可以看出,粒子的位置更新不仅取决于前一粒子的位置,而且取决于粒子的历史位置.粒子的位置更新是长期记忆的结果,可以在一定程度上保护粒子的群体分布和多样性.

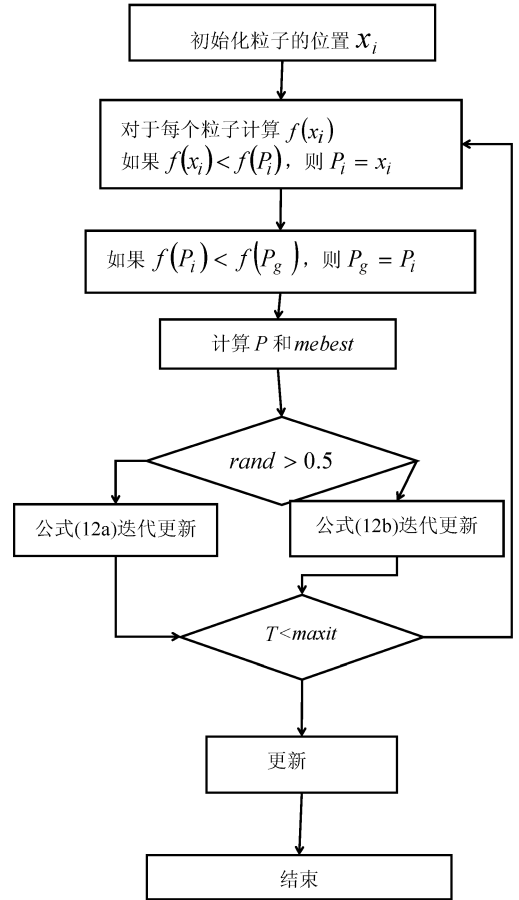


图 2 FQPSO 算法流程图

Fig. 2 The flow chart of FQPSO

4 实验与分析

4.1 测试函数

为了测试本文提出的 FQPSO 算法的性能,对 6 个标准函数^[17]进行了对比实验,这些函数可以分为三组,第一组,单峰函数,函数较为简单,主要用来测试算法的收敛速度;第二组,多峰函数,这类函数有很多个局部极小点,用来测试算法的全局收敛能力和跳出局部最优,避免早熟收敛的能力;第三组,低维函数.表 1 中的搜索范围是指测试函数每一维变量的范围,最优值指的是测试函数的理论最小值.

误差指的是算法优化结果与最优值允许的最大误差,若算法优化结果与最优值之差小于误差,则认为本次优化是成功的.其中 f_1 函数为 Sphere

函数,其广泛用于单峰函数的测试中; f_3 函数为 Rastrigin's 函数,其广泛用于多峰函数的测试中; 而 f_5 函数也常常作为低维函数的标准测试函数.

4.2 对比实验参数设置

为了直观地评价算法的性能,我们分别对分数阶量子粒子群算法取分数阶阶次为 0.1~0.9,步长为 0.1 的 9 个不同阶次的分数阶量子粒子群算法与传统的整数阶算法作对比. 为了公平起见,所有的算法种群量子数均为 30,测试函数的维度均为 30,算法的评价次数均为 3.0×10^4 次,表 2 中的 Mean 代表 50 次结果运行的平均值,Best 为 50 次中的最佳结果.

4.3 实验结果分析

4.3.1 单峰函数对比实验 主要分析算法的收敛速度和收敛精度. 在收敛速度方面,从图 3(a)和图

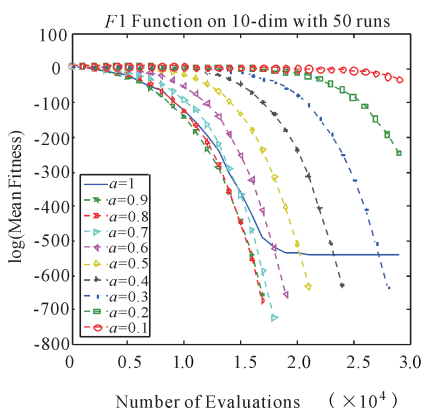
3(b)可以看出,在分数阶阶次大于 0.7 的时候,FQPSO 的收敛速度明显快于 QPSO,由此可见,在单峰函数上不仅保留甚至还增强了标准 QPSO 算法的收敛特性.

在收敛精度方面,从表 2 可以看出,本文提出的 FQPSO 算法在阶次大于 0.2 的时候均取得了最小值,显然在单峰函数的优化上 FQPSO 具有良好的收敛性能. 而阶次在 0.1 和 0.2 时,收敛性不佳的原因是在分数阶微分进行数值阶次过度偏小时,分数阶微分的近似数值解已经远远偏离了该阶次的分数阶微分的真实解. 而在 2 个单峰函数的测试上都能以 100% 的概率达到指定的误差精度. 因此还可以说,FQPSO 在单峰函数上既能保持快速收敛,同时能够大幅度提高收敛精度.

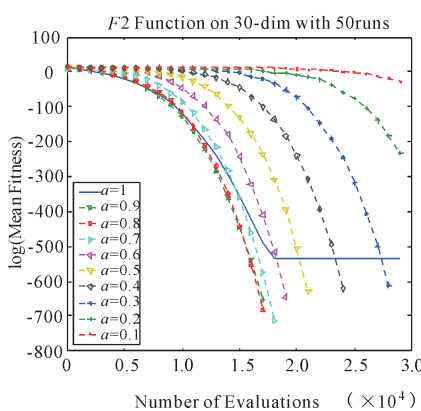
表 1 测试函数

Tab.1 Test function

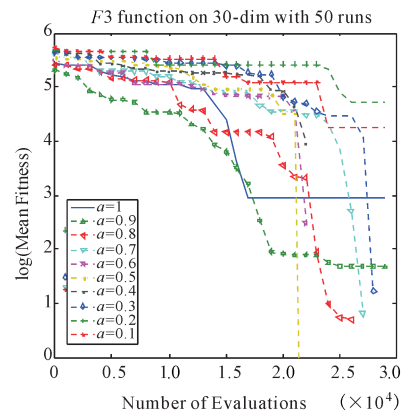
| F | 测试函数 | 搜索范围 | 维数 | 最优值 | 最优解 |
|-------|--|----------------|----|-----|-----|
| f_1 | $\sum_{i=1}^n x_i^2$ | $[-100,100]$ | 30 | 0 | 0 |
| f_2 | $\sum_{i=1}^n (\sum_{j=1}^i x_j)^2$ | $[-100,100]$ | 30 | 0 | 0 |
| f_3 | $\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$ | $[-5.12,5.12]$ | 30 | 0 | 0 |
| f_4 | $\sum_{i=1}^n (\sum_{k=0}^{20} (0.5)^k \cos(2\pi(3)^k(x_i + 0.5))) - n * \sum_{k=0}^{20} ((0.5)^k \cos(2\pi * 3^k * 0.5))$ | $[-5.12,5.12]$ | 30 | 0 | 0 |
| f_5 | $0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2}$ | $[-100,100]$ | 2 | 0 | 0 |
| f_6 | $x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$ | $[-50,50]$ | 2 | 0 | 0 |



(a) f_1



(b) f_2



(c) f_3

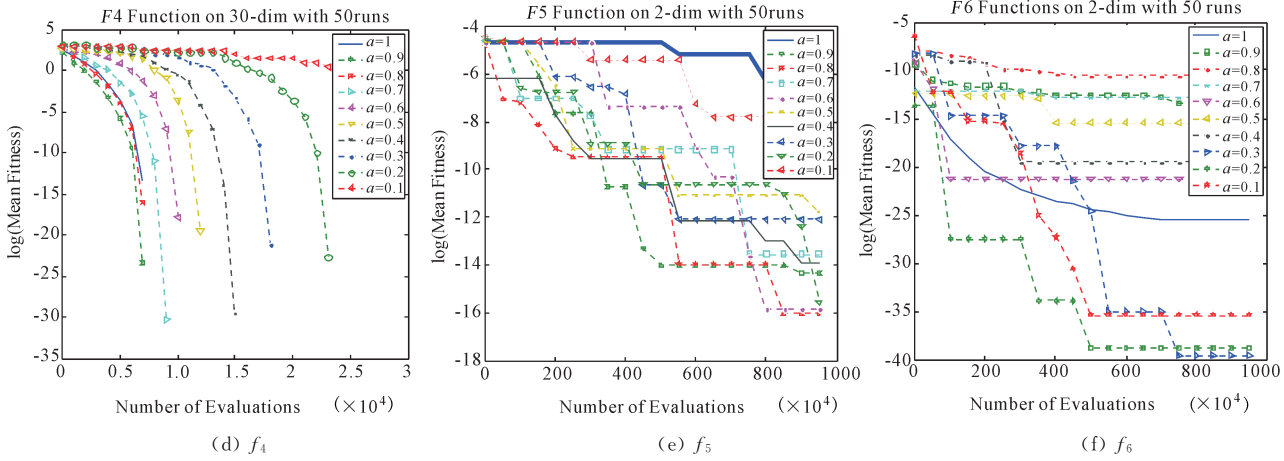


图 3 不同测试函数不同阶次收敛曲线

Fig. 3 Convergence curve of different test functions with different orders

表 2 高维函数对比试验

Tab. 2 High-dimensional function comparison test

| 分数阶次 | | f_1 | | f_2 | | f_3 | | f_4 | |
|--------------|------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | | Dim=10 FEs=10000 | Dim=30 FEs=30000 | Dim=10 FEs=10000 | Dim=30 FEs=30000 | Dim=10 FEs=10000 | Dim=30 FEs=30000 | Dim=10 FEs=10000 | Dim=30 FEs=30000 |
| $\alpha=1$ | Best | 5.6324e-267 | 2.3453e-241 | 3.7568e-258 | 1.7059e-237 | 0.9950 | 11.9395 | 7.1054e-15 | 2.5251e-10 |
| | Mean | 4.3521e-265 | 1.5387e-239 | 1.8379e-255 | 6.8877e-234 | 1.5919 | 16.0250 | 9.9476e-15 | 4.8798e-07 |
| $\alpha=0.9$ | Best | 0 | 0 | 0 | 0 | 1.7764e-15 | 0.9950 | 1.9257e-17 | 5.6302e-11 |
| | Mean | 0 | 0 | 0 | 0 | 0.1090 | 6.6484 | 1.8609e-13 | 3.7460e-08 |
| $\alpha=0.8$ | Best | 0 | 0 | 0 | 0 | 0 | 0.4517 | 4.9016e-25 | 5.4000e-10 |
| | Mean | 0 | 0 | 0 | 0 | 0 | 2.2933 | 5.6360e-18 | 1.1253e-07 |
| $\alpha=0.7$ | Best | 0 | 0 | 0 | 0 | 0 | 0.0297 | 8.3079e-33 | 5.4000e-20 |
| | Mean | 0 | 0 | 0 | 0 | 0 | 2.4696 | 1.1100e-22 | 8.0348e-15 |
| $\alpha=0.6$ | Best | 0 | 0 | 0 | 0 | 0 | 0.5342 | 5.0220e-44 | 3.4005e-12 |
| | Mean | 0 | 0 | 0 | 0 | 0 | 12.4468 | 2.4418e-26 | 1.0209e-08 |
| $\alpha=0.5$ | Best | 0 | 0 | 0 | 0 | 2.456e-08 | 4.5314e-06 | 6.5183e-43 | 1.6486e-11 |
| | Mean | 0 | 0 | 0 | 0 | 0.01413 | 6.4245e-04 | 9.7154e-22 | 2.2779e-09 |
| $\alpha=0.4$ | Best | 0 | 0 | 0 | 0 | 0.1389 | 6.6789 | 4.765e-28 | 3.1412e-17 |
| | Mean | 0 | 0 | 0 | 0 | 1.5126 | 56.6533 | 8.98017e-13 | 3.1149e-14 |
| $\alpha=0.3$ | Best | 0 | 0 | 0 | 0 | 0.3100 | 0.6789 | 2.0456e-16 | 3.9874e-13 |
| | Mean | 0 | 0 | 0 | 0 | 3.3940 | 3.3532 | 1.1142e-14 | 1.0676e-10 |
| $\alpha=0.2$ | Best | 2.1613e-236 | 5.1257e-106 | 2.1552e-152 | 1.0355e-105 | 1.6976e-10 | 10.2080 | 3.1005e-04 | 4.2733e-17 |
| | Mean | 6.6966e-250 | 2.2847e-100 | 7.0445e-145 | 2.2098e-98 | 2.1109e-09 | 185.937 | 6.01e-02 | 2.6818e-12 |
| $\alpha=0.1$ | Best | 2.613e-53 | 3.5677e-16 | 1.4244e-36 | 4.5673e-15 | 1.0415 | 36.4428 | 0.4777 | 0.0813 |
| | Mean | 1.2778e-45 | 3.9080e-12 | 2.0523e-24 | 2.4097e-11 | 16.0174 | 66.9430 | 2.870 | 0.7666 |

4.3.2 多峰函数对比实验 主要测试算法的收敛精度以及跳出局部最优,避免早熟收敛的性能.在收敛速度上:从图 3 的(c),(d)上可以看出,大于 0.8 阶次的 FQPSO 收敛速度明显快于 QPSO,但小于 0.8 阶次的 FQPSO 收敛速度慢于 QPSO.但在收敛精度上除了 0.1 和 0.2 阶次的均优于整数阶.

从表 3 可以看出 FQPSO 的收敛精度明显高于 QPSO,因此,我们可以说由于 FQPSO 具有了分数阶微分长期记忆的特性,FQPSO 跳出局部最优的能力更佳.当然阶次在 0.1 和 0.2 的时候性能依然不够理想,这也是由于阶次过度偏小时,分数阶微分的近似数值解已经远远偏离了该阶次的分数阶微分的真实解造成的.

表 3 低维函数对比试验

Tab. 3 Low-dimensional function comparison test

| 分数阶次 | | f_5 | | f_6 | | | | | |
|--------------|------|-------------------|--------------|-------------------|--------------|------|------------|--------------|------------|
| | | Dim=2 FEs=1000 | | Dim=2 FEs=1000 | | | | | |
| $\alpha=1$ | Best | 3.2129e-08 | $\alpha=0.5$ | 8.4356e-09 | $\alpha=1$ | Best | 3.5034e-12 | $\alpha=0.5$ | 1.0324e-07 |
| | Mean | 0.000819 | | 6.7286e-06 | | Mean | 8.6010e-12 | | |
| $\alpha=0.9$ | Best | 2.5220e-12 | $\alpha=0.4$ | 5.6344e-14 | $\alpha=0.9$ | Best | 8.2342e-09 | $\alpha=0.4$ | 7.3278e-11 |
| | Mean | 7.3016e-07 | | 8.7416e-07 | | Mean | 7.3016e-07 | | |
| $\alpha=0.8$ | Best | 2.1285e-14 | $\alpha=0.3$ | 5.6356e-12 | $\alpha=0.8$ | Best | 3.5459e-07 | $\alpha=0.3$ | 5.0301e-20 |
| | Mean | 1.8083e-07 | | 5.5595e-06 | | Mean | 2.7536e-05 | | |
| $\alpha=0.7$ | Best | 5.3156e-10 | $\alpha=0.2$ | 2.2636e-11 | $\alpha=0.7$ | Best | 2.3498e-07 | $\alpha=0.2$ | 5.1257e-18 |
| | Mean | 9.6610e-07 | | 1.3745e-07 | | Mean | 1.3710e-06 | | |
| $\alpha=0.6$ | Best | 1.3258e-13 | $\alpha=0.1$ | 2.0779e-07 | $\alpha=0.6$ | Best | 5.6432e-13 | $\alpha=0.1$ | 3.5677e-17 |
| | Mean | 1.2437e-07 | | 4.0306e-04 | | Mean | 3.7895e-10 | | |

4.3.3 低维函数对比实验 主要测试算法在维度较低的情况下的收敛速度以及跳出局部最优的能力.从图 3(e)(f)可以看出在分数阶阶次大于 0.5 的时候该算法依然保持着快速收敛的特性,而在收敛精度上,FQPSO 依然保持了快速收敛的特性.

5 结 论

为了更好地提升量子粒子群的全局搜索能力和局部搜索能力,进一步提升算法的性能.本文提出了基于分数阶微分的量子粒子群算法,利用分数阶微积分长期固有的长时记忆的特性和全局性,多重记忆量子空间中粒子的位置,充分利用寻优过程中的历史经验.实验结果表明,本文提出的算法在单峰函数,多峰函数以及低维函数上均有较快的收敛速度和较高的收敛精度,都能够以较高的概率收敛到理论最优.后续我们将研究算法各阶次之间的关系,以及该算法的收敛性.

参考文献:

[1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]// International Symposium on MICRO Machine and Human Science. Nagoya, Japan: IEEE, 2002.

[2] Goldberg D E. Geneticalgorithm in search, optimization, and machine learning[J]. Addition Wesley, 1989, 7: 2104.

[3] Kane B E. A silicon-based nuclear spin quantum computer [J]. Nature, 1998, 393:133.

[4] Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior[C]//Con-

gress on Evolutionary Computation. Portland, OR, USA: IEEE, 2004.

[5] Liu F, Agrawal O P, Momani S, *et al.* Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications[M]. New York: Academic press, 1999.

[6] 郑荟萃, 李森. 一类半线性分数阶微分方程解的存在性 [J]. 四川大学学报:自然科学版, 2017, 54: 29.

[7] 张海燕, 李耀红. 一类高分数阶微分方程积分边值问题的正解[J]. 四川大学学报:自然科学版, 2016, 53: 512.

[8] Oliveira E C D, Machado J A T. A review of definitions for fractional derivatives and integral [J]. Math Probl Engin, 2014, 2014: 1.

[9] Abdelouahab M S, Hamri N E. The grunwald-letnikov fractional order derivative with fixed memory length[J]. Mediterr J Math, 2016, 13:557.

[10] Gao H, Kwong S, Yang J, *et al.* Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation [J]. Inform Sci, 2013, 250:82.

[11] Cheng S C, Wu J N, Tsai M R, *et al.* Fractional phenomena of the spontaneous emission of a two-level atom in photonic crystals[J]. Phys, 2012, 43: 1.

[12] Pires E J S, Machado J A T, Oliveira P B D M, *et al.* Particle swarm optimization with fractional-order velocity[J]. Nonlin Dyn, 2010, 61: 295.

[13] Pires E J S, Machado J A T, De M O P B, *et al.*

- Fractional order dynamics in a particle swarm optimization algorithm [C]// International Conference on Intelligent Systems Design and Applications. Rio De Janeiro, Brazil: IEEE, 2007.
- [14] Chen W N, Zhang J, Lin Y, *et al.* Particle swarm optimization with an aging leader and challengers [J]. IEEE Transact Evolut Comput, 2013, 17: 241.
- [15] 莫才健,武锋强,陈莉,等. 基于量子粒子群算法选择特征的遥感图像分类[J]. 吉林大学学报:理学版, 2018,56:368.
- [16] 朱江,熊加毫,陈红翠. 认知无线网络中基于 HJ-DQPSO 优化的频谱分配机制[J]. 重庆大学学报:自然科学版,2016,28:37.
- [17] Yao X, Liu Y, Lin G M. Evolutionary programming made faster[J]. IEEE Trans Evolut Comput, 2002, 3: 82.

引用本文格式:

中文: 徐来,蒲亦非,周激流. 基于分数阶位置状态的量子粒子群算法[J]. 四川大学学报:自然科学版, 2018, 55: 947.

英文: Xu L, Pu Y F, Zhou J L. An improved quantum particle swarm optimization with fractional-order position [J]. J Sichuan Univ: Nat Sci Ed, 2018, 55: 947.