文章编号:1671-4598(2017)08-0221-05 DOI:10.16526/j.cnki.11—4762/tp.2017.08.057 中图分类号:TP3 文献标识码:A

Logistic 型混合自适应分数阶达尔文粒子群优化算法

荣 兵,陈 华
(中国石油大学(华东)理学院,山东 青岛 266580)

摘要:针对传统的粒子群优化算法中存在的问题及分数阶达尔文微粒群优化(FDPSO)算法收敛速度慢,收敛精度不高的问题,改进其算法中分数阶速度更新策略,同时引入 Logistic 型混合分数阶自适应动态调整策略,得到一种改进的自适应分数阶达尔文粒子群优化(LFDPSO)算法,并通过相应理论分析,证明了该算法在给定条件下的收敛性,并由 6 个经典函数的数值测验表明,Logistic 型混合自适应分数阶达尔文粒子群(LFDPSO)算法在收敛精度和收敛速度上得到了有效改善与提高,粒子在局部最优时的逃逸能力、全局寻优及智能搜索能力显著增强。

关键词:分数阶;粒子群;达尔文;自适应;速度更新;收敛速度

Logistic—model Hybrid Adaptive Fractional Order Darwinian Partical Swarm Optimization Algorithm

Rong Bing, Chen Hua
(College of Science, China University of Petroleum (East China), Qingdao 266580, China)

Abstract: Aiming at the problems existing in the traditional particle swarm optimization algorithm and the problems existing in convergence speed and precision of fractional order Darwin particle swarm optimization (FDPSO) algorithm, improved the fractional order velocity update strategy of the algorithm, at the same time introduce dynamic logistic model hybrid adaptive strategy of the fractional order to form LFDPSO algorithm, through theoretical analysis and prove the convergence of the iterative algorithm under given conditions, and the experiments by six classical test functions show that the LFDPSO algorithm on the convergence accuracy and convergence speed has been further improved and enhanced, the escape ability of particles in local optimum, global optimization and intelligent search ability have achieved effective improvement.

Keywords: fractional order; particle swarm; Darwinian; adaptive; speed—update; convergence rate

0 引言

粒子群优化(PSO, partical swarm optimization)算法是近年来快速发展的一种智能全局寻优方法[1]。这种方法有它不可取代的寻优特点,在优化组合[2]、多目标联合搜索[3]、任务分配[4]、聚类分析[5]、神经网络[6]等众多领域得到了比较广泛的使用。PSO 算法在使用中存在着一些不尽人意的地方,如在寻优的开始阶段,算法的收敛速度较快,而在全局寻优的后期,单纯的粒子群算法容易陷入早熟,进而影响粒子群算法寻优的精度,这也是群智能优化算法的一种通病。

最近几年,为了改进粒子群算法的不足之处,诸多学者做了较为深刻的探索,Jason 等[7]提出达尔文粒子群优化(DPSO, Darwinian partical swarm optimization)算法,将生态环境中的自然选择的策略引入到粒子群算法之中。分数阶微积分理论(FOC, fractional ordercalculus)作为一种快速发展的、非常重要且有用的数学工具在自然科学利于如水文建模,神经网络等领域取得的成果越来越明显[8-9],Pires 等[10]提出了分数阶粒子群优化(FPSO)算法,将分数阶微积分的相关知识引入到粒子群优化算法中,进而,通过改进其速度导数的阶数来控制 FPSO 算法的收敛效率。由 Pires 等所研究的 FPSO 算法中得到启示,Micael 等[11]提出了一种分数阶达尔文粒子群(FDPSO, fractional order Darwinian partical swarm optimization)算法,将分数阶这一数学工具同达尔文粒子群优化算法结合起来,集中起两者的优点,较大程度上提高了收敛速度及收敛精度,实验结果也表明,较之传统的其它粒子群优化算法,该算法其在计算精度和收敛速度上均表现不俗。

本文为了进一步提高其收敛的精度及速度,改进了 FDPSO 算法的分数阶更新公式,但算法在后期任然存在多样性渐失,收敛速度减慢,精度降低,且算法容易陷入局部最优。陈华等[12]提出了基于 logistic 模型之上的一种动态加速因子的自适应微粒群优化(APSO, adaptive partical swarm optimization)算法,该算法能够有效地提高相应算法的收敛速度及收敛精度,尤其是在提高收敛速度方面,有着较好的效果,主要通过把这种 logistic 型的动态调整策略运用到 FDPSO 算法之中形成 Logistic FDPSO 算法,在下文中简称 LFDPSO,且考虑到此模型在算法后期面临的易陷入局部最优的问题,提出了一

收稿日期:2017-02-23; **修回日期**:2017-03-14。
基金项目:国家自然科学基金(41474100);山东省自然科学基金(ZR2013DM015)。
作者简介:荣 兵(1991-),男,山东滨州人,硕士研究生,主要从事应用数学(群智能优化算法、并行算法)方向的研究。
陈 华(1972-),男,山东聊城人,硕士生导师,副教授,主要从事数学领域方向的研究。

种混合的自适应动态调整策略，在后期通过粒子群中粒子本身的状态信息来对分数阶进行调整，使其具有逃逸局部最优的能力，提高其算法的收敛速度和收敛精度。

1 分数阶达尔文粒子群（FDPSO）算法改进

1.1 PSO 算法基本原理

在一个有 N 个粒子所构成的 D 维目标搜索空间的粒子群中，用 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$ 表示第 i 个粒子的位置向量，用 $V_i=(v_{i1},v_{i2},\cdots,v_{iD})$ 表示第 i 个粒子的速度向量。对应的粒子的速度和位置更新公式如下：

$$v_{id}(t+1)=\omega v_{id}(t)+c_1r_{1d}(p_{id}(t)-x_{id}(t))+c_2r_{2d}(p_{gd}(t)-x_{id}(t)) \tag{1}$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \tag{2}$$

在式（1）中，$i=1,2,\cdots N;d=1,2,\cdots,D;\omega$ 是惯性系数，c_1、c_2 是加速系数；r_{1d}、r_{2d} 是 $[0,1]$ 之间的随机数；p_{id} 是第 i 个粒子本身找到的历史最优位置，即局部最优值点；p_{gd} 是该粒子群找到的群体中的最优位置，即全局最优值点。粒子群中每个粒子可以根据式（1）和式（2）进行迭代，最终使得算法逼近全局最优值处。

1.2 DPSO 算法基本原理

DPSO 基本思想是粒子群中每个粒子可以通过找到更好的适应值来获得增加自己的存活寿命的机会，同时可以通过增加新的种群粒子来提高全局寻优过程的后期搜索的速度。如果粒子存活的时间越长，则它产生后代的概率也就越大，而且后代中保存了一些适应值比较好的粒子。在粒子群中，粒子的寿命是否会缩短取决于它在目标搜索过程中有没有找到更好的适应值位置，相应的，如果没能找到延长寿命的较好的适应值，粒子群中那些较差性能的粒子可能就会被删除，当粒子群中粒子的数量减少到一定值时，该粒子群淘汰。

DPSO 算法在设计的时候，专门设置了一个计数器为 SC，它主要记录粒子适应值没有发生相应改变的次数，当它的数值达到我们给定的 $SC_c^{\max}$，就将该粒子从粒子群中删除，而在创建一个新的粒子群时，SC 置为 0。在删除相应的粒子群后，SC 的值并不重置为 0，而是按照式（3）的方式重新设置，使其值接近 $SC_c^{\max}$：

$$SC_c(N_{kill})=SC_c^{\max}\left[1-\frac{1}{N_{kill}+1}\right] \tag{3}$$

其中：N_{kill} 为在粒子适应值没有任何改善的一段时间内粒子群中被删除的粒子的数目。

在该算法的实现过程中，若要产生一个新的粒子群，必须保证粒子群中没有粒子被删除，并且所存在的最大粒子群数量不能超过给定的最大值。综合上述所有要求，该算法创建新的粒子群概率仅仅为：

$$p=f/N_s \tag{4}$$

其中：f 为 [0，1] 之间的随机数，N_s 为粒子群数目。

1.3 FDPSO 算法中分数阶速度更新策略的改进

根据文献[13-16]，可以得出对传统导数的极限定义为：

$$y^{(n)}(t)=\lim_{h\to 0}h^{-n}\sum_{j=0}^{n}(-1)^j\binom{n}{j}y(t-jh),\quad t\in(a,b). \tag{5}$$

其中：$\binom{n}{j}$ 表示二项式系数。

把这个定义扩展到任意的实数 α 就得到了 Grunwald－Letnikov 分数阶导数的定义：

$${}_aD_t^{\alpha}y(t)=\lim_{h\to 0}h^{-\alpha}\sum_{j=0}^{[(t-a)/h]}(-1)^j\binom{\alpha}{j}*y(t-jh),t\in(a,b). \tag{6}$$

即 α 阶 Grunwald－Letnikov 分数阶导数。整理得另一种表达式的形式为：

$${}_aD_t^{\alpha}y(t)=\lim_{h\to 0}h^{-\alpha}*\sum_{j=0}^{[(t-a)/h]}(-1)^j\frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)}*y(t-jh),t\in(a,b). \tag{7}$$

式（7），离散时间的实现表达式可以被定义为：

$${}_aD_t^{\alpha}y(t)=T^{-\alpha}\sum_{j=0}^{r}(-1)^j\frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)}*y(t-jT),t\in(a,b). \tag{8}$$

其中：T 为采样周期，r 为截断阶数。

假定式（1）的惯性权重 $\omega=1$ 以及采样周期 $T=1$，由文献[17]我们可以得到下面的表达式：

$${}_aD_t^{\alpha}v_{id}(t+1)=c_1r_{1d}(p_{id}(t)-x_{id}(t))+c_2r_{2d}(p_{gd}(t)-x_{id}(t)) \tag{9}$$

随着迭代次数的增加，当代粒子群中的粒子与最初几代的关系逐渐淡化，在本文中我们取 $r=4$，即我们只保留前 4 代的速度向量，并取 $T=1$，由式（7）和式（8），可以得到：

$${}_aD_t^{\alpha}v_{id}(t+1)=v_{id}(t+1)-\alpha v_{id}(t)+\frac{1}{2}\alpha(\alpha-1)v_{id}(t-1)-\frac{1}{6}\alpha(\alpha-1)(\alpha-2)v_{id}(t-2)+\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)v_{id}(t-3) \tag{10}$$

再由式（9）和式（10），得改进的速度更新策略如下：

$$v_{id}(t+1)=\alpha v_{id}(t)-\frac{1}{2}\alpha(\alpha-1)v_{id}(t-1)+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)v_{id}(t-2)-\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)v_{id}(t-3)+c_1r_{1d}(p_{id}(t)-x_{id}(t))+c_2r_{2d}(p_{gd}(t)-x_{id}(t)) \tag{11}$$

其形式不同于文献[11，17]等论文中的分数阶速度更新策略：

$$v_{t+1}=\alpha v_t+\frac{1}{2}\alpha v_{t-1}+\frac{1}{6}\alpha(1-\alpha)v_{t-2}+\frac{1}{24}\alpha(1-\alpha)(2-\alpha)v_{t-3}+c_1r_{1d}(p_{id}(t)-x_{id}(t))+c_2r_{2d}(p_{gd}(t)-x_{id}(t)) \tag{12}$$

虽然形式相似，但数值实验结果表明无论在收敛精度还是收敛速度上均有较明显的改善。

2 Logistic 型混合分数阶自适应动态调整策略的引入

传统的调整分数阶 α 的方法是在 [0.5，0.8] 之间线性减小，具体策略为：

$$\alpha=0.8-0.3*\frac{t}{nger} \tag{13}$$

其中：t 为当前的迭代步数，$nger$ 为总的迭代步数。

这种方法采取的是对分数阶的线性减少，在算法的后期，算法陷入局部最优的可能性会变大，为了更好地避免算法的早熟问题的出现，本文提出一种基于 logistic 模型之上的混合调整分数阶 α 的自适应策略。根据粒子最优位置 15 步没有发生较好变化的前后将算法分为前后两个阶段，在每一阶段采取不同的分数阶 α 自适应调整策略。

在算法的第一阶段，分数阶 α 调整策略在算法过程中，为避免其早熟，α 值在第一阶段初期应较大，并且 α 值减小的速度应较大些，能加快算法的收敛速度；在第一阶段后期，α 值减小的速度应趋缓，易搜索局部最优值，使算法趋于稳定。设 α 的最小值和最大值分别为 $\alpha_{\max}$ 和 $\alpha_{\min}$，若迭代开始时 α 的衰减率为 a，随着迭代次数的增加而衰减率减小，当 α 减小到最小值 $\alpha_{\min}$ 时，α 停止减小，即衰减率为零。因此，α 的变化规律符合 logistic 模型[18-19]，即：

$$\begin{cases}\dfrac{\mathrm{d}\alpha}{\mathrm{d}t}=a\left(1-\dfrac{\alpha}{\alpha_{\min}}\right)\alpha\\ \alpha(0)=\alpha_{\max}\end{cases}\tag{14}$$

对式（14）使用分离变量法来求解，可得到相应缩放因子 α 的动态调整公式：

$$\alpha(t)=\frac{\alpha_{\min}}{1+(\dfrac{\alpha_{\min}}{\alpha_{\max}}-1)\exp(-at)}\tag{15}$$

可以通过调整 a 值来调整加速因子 α 的下降速度。给定的初始衰减率 a 值越大些，α 在算法前期的下降速度就越快写，反之，c_1 的下降速度则比较慢。当 $t=0$ 时，$\alpha=\alpha_{\max}$，而当 $t\to+\infty$ 时，比较容易证明 $\alpha=\alpha_{\min}$。

在算法的第二阶段，分数阶 α 值比较小，为了有效避免其早熟现象的发生概率及增大粒子的拓展搜索空间的能力，我们采取一种根据粒子的状态和最优粒子的轨迹信息进行动态调整，具体方法如下：

1）求出粒子群中每个粒子与其他粒子之间距离的和。

$$d_i=\frac{1}{N-1}\sum_{j=1,j\neq i}^{N}\sqrt{\sum_{k=1}^{D}(x_{ik}-x_{jk})^2}\tag{16}$$

2）参照式（14）求出最优粒子与其它粒子的距离之和 d_{gbest}。

3）计算分数阶 α 的阶数。

$$\alpha=\frac{1}{1+e^{-1.3863\Phi}}\tag{17}$$

其中：$\Phi=\left|\dfrac{d_{gbest}-d_{\min}}{d_{\max}-d_{\min}}\right|$，且易知 $\Phi\in[0,1]$，$\alpha\in[0.5,0.8]$。

Logistic 型混合自适应粒子群优化算法（LFDPSO）算法流程如下：

（1）初始化

a）随机生成所有粒子的速度和位置向量；

b）$p_i=X_i$，$i=1,2\cdots,n$；

c）将 $\arg\{\min f(X_i)\}$ 赋给 g_{best} 其中 $f(\cdot)$ 为适应值函数；

（2）终止条件判断

a）如果终止条件已经达到，g_{best} 即为最优解；

b）否则执行步骤 3；

（3）循环 i 从 1 到 n

a）判断粒子处于第几阶段，根据（15）及（17）更新 α；

b）根据式（11）和式（2）更新 V_i 和 X_i；

c）计算第 i 个粒子的适应值；

d）如果粒子群变得更好则奖励粒子群：产生后代粒子、延长粒子寿命；

e）否则惩罚粒子群：销毁粒子群、缩短其寿命；

f）如果 $f(X_i)$ 优于 $f(p_i)$，则 $p_i=X_i$；

（4）结束循环

（5）更新 $g_{best}=\arg\{\min f(p_i)\}$

（6）返回步骤 2。

3　LFDPSO 算法的收敛性分析

设 $\varphi_1=r_{1d}c_1$，$\varphi_2=r_{2d}c_2$，$y=p_{id}(t)$，$\bar{y}=p_{gd}(t)$，则式（11）可以写为：

$$v_{t+1}=\alpha v_t-\frac{1}{2}\alpha(\alpha-1)v_{t-1}+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)v_{t-2}-\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)v_{t-3}+\varphi_1(y-x_t)+\varphi_2(\bar{y}-x_t)\tag{18}$$

由式（2）$x_{t+1}=x_t+v_{t+1}$ 得到 $v_t=x_t-x_{t-1}$，进一步得到：

$$x_{t+1}=x_t+\alpha(x_t-x_{t-1})-\frac{1}{2}\alpha(\alpha-1)(x_{t-1}-x_{t-2})+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)(x_{t-2}-x_{t-3})-\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)(x_{t-3}-x_{t-4})+\varphi_1(y-x_t)+\varphi_2(\bar{y}-x_t)\tag{19}$$

整理得出递推公式：

$$x_{t+1}=(1+\alpha-\varphi_1-\varphi_2)x_t-[\alpha+\frac{1}{2}\alpha(\alpha-1)]x_{t-1}+[\frac{1}{2}\alpha(\alpha-1)+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)]x_{t-2}-[\frac{1}{6}\alpha(\alpha-1)(\alpha-2)+\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)]x_{t-3}+\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)x_{t-4}+\varphi_1 y+\varphi_2\bar{y}\tag{20}$$

式（20）可以用下面的矩阵乘积的形式表示：

$$\begin{bmatrix}x_{t+1}\\x_t\\x_{t-1}\\x_{t-2}\\x_{t-3}\\1\end{bmatrix}=\begin{bmatrix}(1+\alpha-\varphi_1-\varphi_2) & [\alpha+\frac{1}{2}\alpha(\alpha-1)] & [\frac{1}{2}\alpha(\alpha-1)+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)] & [\frac{1}{6}\alpha(\alpha-1)(\alpha-2)+\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)] & \frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3) & \varphi_1 y+\varphi_2 y\\ 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_t\\x_{t-1}\\x_{t-2}\\x_{t-3}\\x_{t-4}\\1\end{bmatrix}$$

上面矩阵的特征多项式为：

$$(\lambda-1)[\lambda^5-\lambda^4(1+\alpha-\varphi_1-\varphi_2)+\lambda^3[\alpha+\frac{1}{2}\alpha(\alpha-1)]-\lambda^2[\frac{1}{2}\alpha(\alpha-1)+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)]+\lambda[\frac{1}{6}\alpha(\alpha-1)(\alpha-2)+\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)]-\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)] \tag{21}$$

该特征多项式的一个根为 $\lambda_1=1.0$，设其他的特征根分别为 $\lambda_2,\lambda_3,\lambda_4,\lambda_5,\lambda_6$，则式（20）的当前关系式为：

$$x_t=k_1+k_2\lambda_2^t+k_3\lambda_3^t+\cdots+k_6\lambda_6^t \tag{22}$$

其中：k_1,k_2,k_3,k_4,k_5,k_6 均是常数。

下证 x_t 的收敛性，即：

$$\lim_{t\to\infty}x_t=\lim_{t\to\infty}k_1+k_2\lambda_2^t+k_3\lambda_3^t+\cdots+k_6\lambda_6^t \tag{23}$$

显然地，只需证 $\lambda_2,\lambda_3,\lambda_4,\lambda_5,\lambda_6$ 的模都小于 1。下面分析特征多项式（21），由 $\alpha\in[0.5,0.8]$，r_{1d}、$r_{2d}\in[0.3,0.6]$（本文在后面数值试验中随机数的选取皆在此区间内，下面可证明在此区间内选取随机数能够有效保证算法的收敛），$3\leqslant c_1+c_2\leqslant 4$；以及 $\varphi_1=r_{1d}c_1$，$\varphi_2=r_{2d}c_2$；可以进一步得到下面的不等式：$-0.9\leqslant(1+\alpha-\varphi_1-\varphi_2)\leqslant 0.9$；$\frac{1}{2}\alpha(\alpha-1)$ 的最大值为 -0.08，最小值为 -0.125；$\frac{1}{6}\alpha(\alpha-1)(\alpha-2)$ 的最大值为 0.0625，最小值为 0.032。下证当 $|\lambda|>1$ 时，特征多项式无解即可。

当 $\lambda>1$ 时，特征多项式中 $(\lambda-1)$ 恒正，如果此时另一部分，此处我们记为 $f(\lambda)$ 恒正则证毕，当 $\lambda<1$ 时，特征多项式中 $(\lambda-1)$ 恒负，此时 $f(\lambda)$ 如果恒负则证毕，由前面的条件容易求出 $f(\lambda)$ 当 $\lambda=1$ 时大于 0，当 $\lambda=-1$ 时小于 0，对 $f(\lambda)$ 求导得：

$$f'(\lambda)=5\lambda^4-4\lambda^3(1+\alpha-\varphi_1-\varphi_2)+3\lambda^2[\alpha+\frac{1}{2}\alpha(\alpha-1)]-2\lambda^1[\frac{1}{2}\alpha(\alpha-1)+\frac{1}{6}\alpha(\alpha-1)(\alpha-2)]+[\frac{1}{6}\alpha(\alpha-1)(\alpha-2)+\frac{1}{24}\alpha(\alpha-1)(\alpha-2)(\alpha-3)] \tag{24}$$

由前面的条件容易求出当 $|\lambda|>1$ 时 $f'(\lambda)$ 恒大于 0，故当 $\lambda>1$ 时 $f(\lambda)>f(1)>0$，当 $\lambda<1$ 时 $f(\lambda)<f(-1)<0$，所以当 $|\lambda|>1$ 时，特征多项式恒大于 0，故不存在比 1 大的特征根，证毕，并且对于所有的截断 r 都成立。

4 数值实验及结果分析

为了验证 Logistic 型混合自适应粒子群优化（LFDPSO）算法的较之传统的分数阶粒子群优化（FO－PSO）算法，达尔文粒子群优化（DPSO）算法，分数阶达尔文粒子群（FDPSO）算法的性能，选取该 6 个典型的实验函数进行测试，它们是在群智能优化算法中被广泛采用的测试函数（求最小值），即 Sphere、Rosenbrock、DeJong F4、Rastrigin、Griewank 和 Ackley 函数[20]，其表达式为：

Sphere 函数：

$$f_1(x)=\sum_{i=1}^{D}x_i^2 \tag{25}$$

其中：$x_i\in[-50,50],i=\{1,2,\cdots,D\},f(x^*)=0.0$。

Rosenbrock 函数：

$$f_2(x)=\sum_{i=1}^{D}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2) \tag{26}$$

其中：$x_i\in[-100,100],i=\{1,2,\cdots,D\},f(x^*)=0.0$。

DeJong F4 函数：

$$f_3(x)=\sum_{i=1}^{D}x_i^4 \tag{27}$$

其中：$x_i\in[-20,20],i=\{1,2,\cdots,D\},f(x^*)=0.0$。

Rastrigin 函数：

$$f_4(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10) \tag{28}$$

其中：$x_i\in[-5.12,5.12],i=\{1,2,\cdots,D\},f(x^*)=0.0$。

Griewank 函数：

$$f_5(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1 \tag{29}$$

其中：$x_i\in[-600,600],i=\{1,2,\cdots,D\},f(x^*)=0.0$。

Ackley 函数：

$$f_6(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e \tag{30}$$

其中：$x_i\in[-32,32],i=\{1,2,\cdots,D\},f(x^*)=0.0$。

这 6 个函数中前 3 个为单峰函数，后三个为多峰函数，它们的全局最优值为 $f(x^*)=0.0$。本章试验中的算法均选取相同参数 r_{1d}、$r_{2d}\in[0.3,0.6]$，$\omega=1$，$c_1=1.5$，$c_2=1.5$，其中除 LFDPSO 算法外的分数阶 α 采取的是线性减小，具体策略如式（12）。

其数值实验结果如图 1 所示，通过对比可以得到在迭代步数相同的情况下（本文为迭代 200 步），改进后的 LFDPSO 算法其最终收敛精度明显高于分数阶粒子群优化（FO－PSO）算法，达尔文粒子群优化（DPSO）算法，分数阶达尔文粒子群（FDPSO）算法，在收敛速度方面，尤其在迭代后期收敛的速度的改善较为明显，且有比较知道算法在后期具备了逃逸局部最优的能力，全局收敛能力及智能搜索能力加强。

5 结束语

分数阶导数的应用对于改进粒子群算法来说具有非常重要的意义，使得粒子群算法中局部和全局速度特点更容易被算法理解，换句话说，即为粒子群算法更加智能化。

本文通过改进 FDPSO 算法中分数阶速度更新策略，将分数阶更好的融入到 PSO 算法中来，充分利用了粒子群在寻优过程中的历史信息，并采取了 Logistic 型混合的分数阶自适应动态调整策略，使得再改善收敛精度及收敛速度的基础上，提高了粒子扩展搜索空间的能力，通过实验验证，LFDPSO 算法达到了较好的效果，但是在求解某些多峰函数的时候结果仍有陷入局部最优的情况，在陷入局部最优时，粒子如何逃逸仍然是下面一段时期的工作重点，在下一步工作中，将惯性权重因子及加速因子的调整策略与我们的方法结合，寻找效果更好的方法，并将其应用到电磁计算如随钻侧井的参数反演问题中，加快问题求解速度及提高问题的精度。

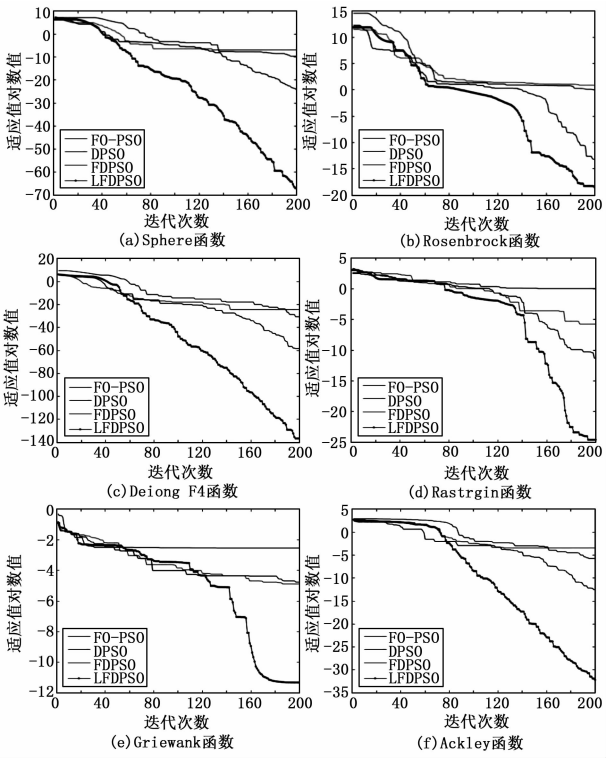

图　LFDPSO 算法在不同的测试函数上的性能比较

参考文献：

[1] Kennedy J，Eberhart R. Particle swarm optimization [A]. Proc IEEE International Conf on Neural Networks [C]. 1995：1942－1948.

[2] Chen W N，Zhang. A novel set－based particle swarm optimization method for discrete optimization problem [J]. IEEE Transactions on Evolutionary Computation，2010，14 (2)：278－300.

[3] Zhang T，Hu T S，Zheng Y，et al. An improved particle swarm optimization for solving believed mulitiobjective programming problem [J]. Journal of Applied Mathematics，2012，2 (4)：1－13.

[4] Ho S Y，Lin H S，Liauh W H，et al. OPSO：Orthogonal particle swarm optimization and its application to task assignment problems [J]. IEEE Transactions on Systems，Man，Cybernetics A：Systems，Humans，2008，38 (2)：288－298.

[5] Nie F，Tu T，Pan M，et al. K－Harmonic means data clusteing with PSO Algorithm [M]. Springer Berlin Heidelberg，2012：67－73.

[6] Varshney S，Srivastava L，Pandit M. Parameter tuning of statcom using particle swarm optimization based neural network [J]. Intelligent and Soft Computing，2012，130 (3)：813－824.

[7] Tillett J，Rao T，Sahin F，et al. Darwinian particle swarm optimization [A]. Indian International Conference on Artificial Intelligence [C]. 2005：1474－1487.

[8] Gutierrez R E，Rosario J M，Machado J T. Fractional order calculus：basic concepts and engineering applications [J]. Mathematical Proble－ ms in Engineering，2010，2010：1－19

[9] Machado J A T，Jesus I S，Barbosa R，et al. Application of fractional calculus in engineering [J]. Dynamics，Games and Science I，Springer Proceedings in Mathematics，2011，1：619－629.

[10] Pires J A，Moura P B，Oliveir A M，et al. Particle swarm optimization with fractional－order velocity [J]. Nonlinear Dynamics，2010，61 (1－2)：295－301.

[11] Couceiro M S，Rocha R P，Fonseca F N M，et al. Introducing the fractional － order Darwinian PSO [J]. Signal，Image and Video Processing，2012，6 (3)：343－350.

[12] 陈华，范宜仁，等. 一种动态加速因子的自适应微粒群优化算法 [J]. 中国石油大学学报，2010，34 (6)：173－176.

[13] Podlubny I. Fractional differential equations [M]. San Diego：Academic Iess，1999.

[14] Lshehawey E，Elbarbary E F，et al. On the solution of the endolymph equation using fractional calculus [J]. Appl. Math. Comput.，2001，124 (3)：337－341.

[15] Camargo R F，Chiacchio A O，Oliveira E C. Differentiation to fractional orders and the fractional telegraph equation [J]. Math. Phys.，2008，49 (3)：033－505.

[16] Diethelm K. The analysis of fractional differential equations [M]. Berlin：Springcr－Vd：lag，2010.

[17] Pires E J S，Machado J A T，Oliveira P B M，Cunha，et al. Particle swarm optimization with fractional－order velocity [J]. Nonlinear Dyn，2010，61 (1/2)：295－301.

[18] Munkres J R. Topology [M]. 2nd ed. London：Prentice－Hall Inc，2000.

[19] Guez－Lopez J S R，Romaguera S. The relationship between the Vietoris topology and the Hausdorff quasiuniformity [J]. Topology and Its Applications，2002，124：451－464.

[20] 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.

(上接第 220 页)

[3] 陈志华，刘晓勇. 云计算下大数据非结构的稳定性检索方法 [J]. 现代电子技术，2016，39 (6)：58－61.

[4] 王跃飞，于　炯，鲁　亮. 面向内存云的数据块索引方法 [J]. 计算机应用，2016，36 (5)：1222－1227.

[5] 张博雅，胡晓辉. 一种基于全域子空间分解挖掘的 QoS 准确预测方法 [J]. 计算机科学，2014，41 (1)：217－224.

[6] 方　芳，程效军. 海量散乱点云快速压缩算法 [J]. 武汉大学学报 (信息科学版)，2013，38 (11)：1353－1357.

[7] 张景祥，王士同，邓赵红，等. 融合异构特征的子空间迁移学习算法 [J]. 自动化学报，2014，40 (2)：236－246.

[8] 张　涛，唐振民，吕建勇. 一种基于低秩表示的子空间聚类改进算法 [J]. 电子与信息学报，2016，38 (11)：2811－2818.

[9] 邓志刚，曾国荪，谭云兰，等. 云存储内容分发网络中的能耗优化方法 [J]. 计算机应用，2016，36 (6)：1515－1519.

[10] 张　盛，鄢　傲，向忠胜，等. 基于全网能量均衡的 WirelessHART 图路由算法 [J]. 计算机应用研究，2014，31 (5)：15201523.

[11] Mahboubi H，Moezzi K，Aghdam A G，et al. Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors [J]. IEEE Transactions on Industrial Informatics，2014，10 (1)：163－174.

[12] Mahboubi H. Distributed deployment algorithms for efficient coverage in a network of mobile sensors with nonidentical sensing Capabilities [J]. IEEE Transactions on Vehicular Technology，2014，23 (8)：3998－4016.