

# 基于 Lévy 变异的微粒群算法<sup>①</sup>

林振思<sup>1,2</sup>, 张岐山<sup>1</sup>

<sup>1</sup>(福州大学 经济与管理学院, 福州 350116)

<sup>2</sup>(福建工程学院 管理学院, 福州 350118)

**摘要:** 微粒群算法因其实现简单及优化效果较好而得到广泛应用, 但也存在易早熟和局部收敛的缺点; 结合 Lévy 飞行的特性, 提出了一种新的带 Lévy 变异的微粒群算法, 并对其收敛性进行分析, 指出该算法依概率收敛于全局最优解. 通过对 8 个标准测试函数的仿真实验, 结果表明改进算法中的 Lévy 变异能够利用粒子的当前知识并增加群体的多样性, 从而能够更有效地平衡局部搜索和全局搜索, 使其具有更好的性能, 最后对改进算法的各参数设置进行了探讨分析.

**关键词:** Lévy 分布; 微粒群算法; 变异; 多样性

## Particle Swarm Optimization Based on Lévy Mutation

LIN Zhen-Si<sup>1,2</sup>, ZHANG Qi-Shan<sup>1</sup>

<sup>1</sup>(School of Economics and Management, Fuzhou University, Fuzhou 350116, China)

<sup>2</sup>(School of Management, Fujian University of Technology, Fuzhou 350118, China)

**Abstract:** Particle swarm optimization (PSO) was applied in many fields because of its simplicity and fast convergence, but it is easily prone to be premature and get stuck in local optima. Combination with the characteristics of Lévy flight, this paper proposes a new variation of PSO with Lévy mutation (LévyPSO), and then analyzed its convergence and pointed out that the algorithm convergence in probability for the global optima. The experiments is conducted on 8 classic benchmark functions, the results show that the Lévy mutation can use the current knowledge of particles and increase the diversity of population. Thus, the proposed algorithm has better performance because of it can more effectively balance the global search and local search. The parameters settings of the proposed algorithm are discussed in the final.

**Key words:** Lévy distribution; particle swarm optimization; mutation; diversity

Kennedy 和 Eberhart 于 1995 年提出了一种基于种群搜索的微粒群算法(Particle Swarm Optimization, 简称 PSO)<sup>[1]</sup>, 该算法用组织的社会行为代替进化计算的自然选择机制, 通过种群中各个体之间的协作来实现对问题解空间的搜索. 在此基础上, Shi 和 Eberhart 提出了 PSO 算法的惯性权重模型, 其位置和速度更新方式如式(1)和式(2)所示<sup>[2]</sup>.

$$v_i^{t+1} = wv_i^t + r_1c_1(p_i^t - x_i^t) + r_2c_2(p_g^t - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

其中,  $v_i^t$  和  $x_i^t$  分别表示第  $i$  个粒子在  $t$  时刻的速度和

位置,  $p_i^t$  表示第  $i$  个粒子的历史最优位置,  $p_g^t$  是整个种群的历史最优位置,  $r_1$  和  $r_2$  是 [0,1] 之间的随机数,  $c_1$  和  $c_2$  分别是个体认知系数和社会认知系数.

由于实现简单、参数较少、优化效果好, 在电力调度<sup>[3]</sup>、工业控制<sup>[4]</sup>、机械加工<sup>[5]</sup>、模式识别<sup>[6]</sup>、经济调度<sup>[7]</sup>等领域得到了广泛的应用. 但是 PSO 算法也存在着早熟收敛的问题, 随着进化代数增加, 种群的多样性迅速下降, 算法容易陷于局部极值点. 为了避免 PSO 算法的早熟收敛, 可以通过种群重新初始化<sup>[8]</sup>、改变种群的领域结构<sup>[9]</sup>等方法来增强种群的多样性; 也

① 基金项目:国家自然科学基金(61300104);福建省自然科学基金(2013J01230);福建软科学项目(2013R0057)

收稿时间:2016-02-17;收到修改稿时间:2016-04-05 [doi:10.15888/j.cnki.csa.005426]

可以通过引入遗传变异<sup>[10]</sup>、量子旋转门<sup>[11]</sup>、小波变换<sup>[12]</sup>、差分算子<sup>[13]</sup>、随机变异等算子来使得粒子跳出局部极值点,增强全局搜索能力.其中随机变异算子思想简单且易于实现,研究的成果较多:Higashi等在每次迭代中对粒子的位置以一定的概率进行高斯变异操作<sup>[14]</sup>,但随机选择不能保证对“差”的粒子进行变异;Krohling等将高斯变异和柯西变异<sup>[15]</sup>、指数概率分布变异算子<sup>[16]</sup>、重新初始化(均匀变异)<sup>[17]</sup>等随机变异算子应用于GPSO和BBPSO算法,但粒子连续没有更新的概率较低使得变异频度小,从而影响改进效果;赫然等定义了一种“逃逸运动”的变异算子,当粒子的速度小于一定阈值(动态调整)时,将粒子的 $v_i$ 重置为一个随机初始值<sup>[18]</sup>,但频繁的变异降低了局部搜索能力;Li等在每次迭代中对每个粒子进行柯西变异操作,通过自然选择机制保留更优的粒子<sup>[19]</sup>,但同时 $V$ 和 $X$ 进行变异,因两者相互关联而会导致群体混乱;Wang等对全局最优粒子进行柯西变异操作,以引导其他粒子飞往更优的位置<sup>[20]</sup>,在此基础上又提出了自适应柯西变异算子,并同时 $p_i$ 进行变异<sup>[21]</sup>,Jana和Sil在对随机选择的粒子进行Lévy变异以增加群体的多样性的同时,也对 $p_g$ 进行多项式变异使其跳出局部极值点<sup>[22]</sup>,但对 $p_g$ 的变异不能保证往最优的方向飞行,有可能误导其他粒子的搜索方向.以上这些工作在一定程度上提高了PSO算法的性能,但是在一些复杂问题上的效率和性能还有待提高.

综合来看,引入随机变异算子来改进PSO算法要解决三个问题:(1)选择准则—即每次迭代选择哪些粒子进行变异;(2)算子选择—即要选择哪种变异算子,如高斯变异、均匀变异等;(3)对象选择—即变异算子要对粒子的速度 $V$ 还是位置 $X$ 进行扰动.本文结合随机变异思想和Lévy飞行给出改进的PSO算法,先定义了粒子的停滞状态,然后对连续停滞代数超过一定阈值的粒子的位置进行Lévy变异.通过对8个经典的测试函数进行实验计算并与其它算法对比,结果显示改进算法在各测试函数上都有很好的表现,并对改进算法的参数选择进行了讨论.

## 1 微粒群算法的改进

### 1.1 Lévy分布

Lévy飞行是一种短距离蹦跳式的探索性搜索与偶尔较长距离的开发性搜索相间的搜索策略.

Reynolds的研究表明多种生物的觅食行为都符合或近似于Lévy飞行搜索,且当在食物来源分散且随机的环境中,Lévy飞行是最理想的搜索策略<sup>[23]</sup>.Lévy飞行服从Lévy稳定分布,没有统一的概率密度函数.Mantegna提出了一种快速产生Lévy随机数的仿真算法,通过如式(3)的非线性变换产生一个Lévy随机数<sup>[24]</sup>.

$$L(\alpha, \gamma) = \gamma^{1/\alpha} [(K(\alpha) - 1)e^{-\gamma/C(\alpha)} + 1] \quad (3)$$

其中

$$v = \frac{x}{|y|^{1/\alpha}} \quad (4)$$

式(3)中, $\gamma$ 是尺度参数,用来衡量分布的宽度,一般可以取 $\gamma=1$ ; $\alpha \in (0,2]$ 是特征指数,用来控制分布的峰度;式(4)中, $x, y$ 是服从标准差为 $\sigma_x, \sigma_y$ 的正态分布的随机变量;不同 $\alpha$ 水平的 $\sigma_x, \sigma_y, K(\alpha), C(\alpha)$ 的取值在文献[24]中给出.当 $\alpha=2$ 和 $\alpha=1$ 时,Lévy分布则转化为Gaussian分布和Cauchy分布.

### 1.2 带Lévy变异的微粒群算法

PSO算法中群体的多样性随着进化代数的增加迅速下降,从而使得算法易早熟或局部收敛,通过对“劣”粒子进行变异可增加群体的多样性从而提高算法的性能.

当一个粒子在一次迭代中没有更新它的 $p_i$ ,则称该粒子停滞;如果粒子在连续的若干次迭代中停滞,说明该粒子可能已陷入局部极值点,是一个“劣”粒子.因此改进算法的思想是:当一个粒子连续停滞的代数超过最大允许停滞代数(Max Stagnation Iterations, MSI),则对该粒子的位置 $x_i$ 按式(5)进行变异扰动,即粒子的 $x_i$ 各维加上一个服从Lévy分布的随机数.

$$x'_{id} = x_{id} + \lambda \cdot \text{LévyRand}() \cdot x_{\max} \quad (5)$$

其中: $\lambda$ 称为变异尺度参数;LévyRand()是服从Lévy分布的随机数; $x_{\max}$ 为搜索领域的上边界.这种带Lévy变异的PSO算法称之为LévyPSO算法(Particle Swarm Optimization with Lévy Mutation, LévyPSO).

LévyPSO算法通过人为机制和随机机制来平衡全局的开发搜索和局部探索搜索.在改进算法中设置最大允许停滞代数MSI可以控制粒子变异模式,在算法进化的初始阶段粒子能够较容易的更新 $p_i$ 使得变异发生较少,随着算法的继续迭代,粒子更新 $p_i$ 的频率减少而变异操作增加,这是平衡开发和探索的人为机制.因为Lévy分布会产生一些小的随机数,使得粒子能够利用已有的信息进行小范围的探索式搜索,又偶

尔会产生一个较大的随机数,使得粒子能够跳出局部极值点进行更大范围的开发性搜索,这是追求全局开发与局部探索平衡的随机机制;通过随机机制和人为机制的结合,使得 LévyPSO 算法的性能得到明显的提高。

### 1.3 LévyPSO 的收敛性分析

对于随机搜索算法, Solis 和 Wets 给出了其收敛的假设条件和定理证明<sup>[25]</sup>,相关的主要假设和定理如下。

假设 1.  $f(D(x,\xi)) \leq f(x)$  且如果  $\xi \in S$ ,  $f(D(x,\xi)) \leq \min\{f(x), f(\xi)\}$ 。

假设 2.  $\forall A \subseteq S, v(A) > 0$ , 有  $\sum_{k=0}^{\infty} [1 - u_k(A)] = 0$ 。

其中,  $D$  为产生问题解的函数,  $f$  为目标函数,  $S \subseteq R^n$  是问题的解空间,  $\xi$  是从样本空间  $(R^n, B, u)$  产生的随机向量,  $v(A)$  是  $A$  的 Lebesgue 测度,  $u_k(A)$  由式(6)给出,表示在第  $k$  步产生的解  $x^k$  落在  $A$  的概率。

$$u_k(A) = P(x^k \in A | x^0, x^1, \dots, x^{k-1}) \quad (6)$$

可见,假设 1 要求搜索算法产生的目标函数值序列  $\{f(x^k)\}_{k=0}^{\infty}$  是单调递减的;假设 2 则要求任给  $S$  中的一个非空子集  $A$ ,当  $k \rightarrow \infty$  时,  $x^k$  落入  $A$  的概率为 1,即当搜索算法无限进行时其产生的解遗漏  $A$  的概率为 0,或者说搜索空间要完全覆盖  $S$ 。

定理 1. 假设  $f$  函数是可测的,  $S \subseteq R^n$  是可测的,并且满足假设 1 和假设 2. 令  $\{x^k\}_{k=0}^{\infty}$  是算法产生的序列,则

$$\lim_{k \rightarrow \infty} p[x^k \in R_{\varepsilon, M}] = 1$$

其中,  $R_{\varepsilon, M}$  是问题的最优区域,对于连续问题即为最优解,  $P[x^k \in R_{\varepsilon, M}]$  表示在第  $k$  步  $x^k$  落在  $R_{\varepsilon, M}$  的概率。该定理说明了如果一个随机搜索算法满足以上两个假设条件,则该随机搜索算法以概率 1 收敛于全局最优解。

LévyPSO 是集群体协作和随机变异为一体的随机搜索算法,其进化的最优解序列为  $\{p_g^t\}_{t=0}^{\infty}$ ,  $t$  为进化代数,  $p_g^t$  是第  $t$  代时的全局最好位置。LévyPSO 每次迭代产生最优解的函数定义如式(7)所示,容易看出该算法产生的每代的最优适应度值序列  $\{f(p_g^t)\}_{k=0}^{\infty}$  是单调递减的,故满足假设 1。

$$D(p_g^{t-1}, x_i^t) = \begin{cases} p_g^{t-1}, f(p_g^{t-1}) \leq \min_i f(x_i^t) \\ x_i^t, f(p_g^{t-1}) > \min_i f(x_i^t) \end{cases} \quad (7)$$

LévyPSO 算法的搜索过程是在 PSO 搜索中嵌入 Lévy 随机搜索,也可以看成是在 Lévy 随机搜索中加入 PSO 搜索。为了便于分析,这里将 LévyPSO 简化成一个 Lévy 随机搜索算法,即群体中的每个粒子是由式(5)变异后进行一定步骤的 PSO 搜索。而 Lévy 分布的支撑集为  $R$ ,因此由式(5)产生的随机空间  $L$  是以  $x_i^t$  为中心,半径  $r = \infty$  的超球体,有

$$\text{diam}(L) = \infty > 2\text{diam}(S) \rightarrow L \supset S$$

算法中又对  $x_i$  进行了规范处理  $g(x_i)$ ,以保证各分量不超出问题的解空间  $S$ ,记规范处理为  $g: L \rightarrow L'$ 。

$$g(x_i^t) : x_{id} = \begin{cases} x_{\max}, x_{id} > x_{\max} \\ -x_{\max}, x_{id} < -x_{\max} \\ x_{id}, \text{否则} \end{cases} \quad (8)$$

其中  $x_{id}^t$  表示第  $i$  个粒子第  $t$  代的第  $d$  维分量,  $x_{\max}$  和  $-x_{\max}$  分别表示  $S$  的上界和下界。显然规范处理后的  $L'=S$ ,即 LévyPSO 算法的搜索空间完全覆盖  $S$ ,因此满足假设 2。根据定理 1 可知 LévyPSO 算法依概率收敛于全局最优解。

## 2 仿真实验设置

### 2.1 测试函数

本文选择了常用的 8 个经典测试函数进行数值实验<sup>[18,22,26]</sup>,其数式和搜索区间如表 1 所示。前 4 个是单模态函数,其中 F1 和 F2 函数通常用来测试算法的寻优速度;F3 函数因缺乏搜索方向很难找到全局最优值,常用来测试算法的综合性能;F4 函数包含有随机噪声;后 4 个测试函数是多模态函数,其中 F5 和 F6 函数的极值呈规律分布,相对较容易搜索到全局极值点;F7 函数的全局极值点外包围着类似等高线的局部极值环,这些环易使算法陷入局部极值。F8 函数的表面粗糙,各极值点间距离较远,极易使优化算法陷入局部极值,是测试算法全局收敛的经典函数。F8 函数的最优值为 -12569.5,其他函数的最优值均为 0。

表 1 测试函数

函数	函数式	搜索区间
F1	$10^6 x_1^2 + \sum_{i=2}^n x_i^2$	[-100,100]
F2	$\sum_{i=1}^n [\sum_{j=1}^i x_j]^2$	[-100,100]
F3	$\sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	[-50,50]
F4	$\sum_{i=1}^n ix_i^4 + \text{random}[0,)$	[-1.28,1.28]

F5	$\sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-300,300]
F6	$\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]
F7	$\sum_{i=1}^{n-1} [\sqrt[4]{x_i^2 + x_{i+1}^2} (\sin(50\sqrt{x_i^2 + x_{i+1}^2}) + 1)]$	[-100,100]
F8	$\sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })]$	[-500,500]

2.2 参数设置

所有实验的种群规模均为 20, 粒子的维度设置为 30, 每个测试函数各独立运行 50 次, 每次迭代进化 4000 代. 为了对不同的随机变异算子进行实验比较, GaussianPSO 算法、CauchyPSO 算法和 UniformPSO 算法分别引入了高斯变异算子、柯西变异算子和均匀变异算子. 各算法的实验参数设置如表 2 所示.

表 2 各算法的实验参数设置

算法	wmax	wmin	c1	c2	$\lambda$	MSI	$\alpha$	$\gamma$
PSO	0.9	0.4	2.0	2.0	-	-	-	-
LévyPSO	0.9	0.4	2.0	2.0	1	10	1.5	1
GaussianPSO	0.9	0.4	2.0	2.0	1	10	2	1
CauchyPSO	0.9	0.4	2.0	2.0	1	10	1	1
UniformPSO	0.9	0.4	2.0	2.0	1	10	-	-

3 仿真结果分析

3.1 LévyPSO 与 PSO 比较

表 3 是 LévyPSO 算法与 PSO 算法的运行结果比较. 表中的数据是各算法在 8 个函数上分别运行 50 次的最优值的最小值(Min)、平均值(Mean) 和标准差(Std. Dev.). 可以看到 LévyPSO 算法在 8 个函数上的运算结果比 PSO 算法有很大的改进, 在 F1、F2、F5、F6 和 F7 这 5 个函数的 50 次运行中都能收敛到全局最优值(在 Python 中小于 1E-308 时显示为 0), 在 F3、F4、F8 这三个函数中所得到的结果法非常接近全局极值.

表 3 LévyPSO 与 PSO 实验结果

函数	算法	Min	Mean	Std. Dev.
F1	PSO	0	9609.01	10487.38
	LévyPSO	0	0	0
F2	PSO	150.65	8821.75	8544.78
	LévyPSO	0	0	0
F3	PSO	28.40	266.19	1446.65
	LévyPSO	0	8.98E-06	3.03E-05
F4	PSO	0.00013	2.41	6.32
	LévyPSO	1.47E-06	1.64E-04	1.52E-04

F5	PSO	0	26.39	49.12
	LévyPSO	0	0	0
F6	PSO	0	57.56	29.66
	LévyPSO	0	0	0
F7	PSO	0	7.28	10.52
	LévyPSO	0	0	0
F8	PSO	-11374.80	-8403.83	1157.02
	LévyPSO	-12569.49	-11483.63	1091.87

图 1 和图 2 分别是 LévyPSO 与 PSO 在 F1 和 F2 函数上各自最优的一次进化曲线, 图中因取对数的需要, 作图时将最优值加上一个很小的常量. 可以看出 LévyPSO 收敛速度明显快于 PSO. 图 3 和图 4 是 F3 和 F8 这两个函数的各自最优的一次进化曲线. 图中显示在 F3 函数上 PSO 陷入早熟收敛, 而 LévyPSO 在进化后期还能够改进搜索结果; 在 F8 函数上 PSO 陷入了局部收敛, 而 LévyPSO 则能够全局收敛. 可见 LévyPSO 的全局搜索能力有显著提高. 因此, LévyPSO 能够更好的平衡局部搜索和全局搜索, 在搜索速度和全局搜索能力上都强于 PSO.

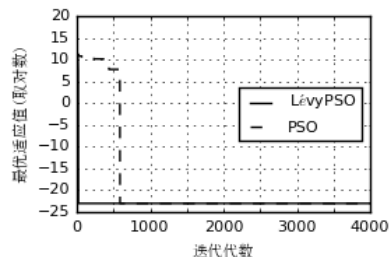


图 1 F1 函数进化曲线

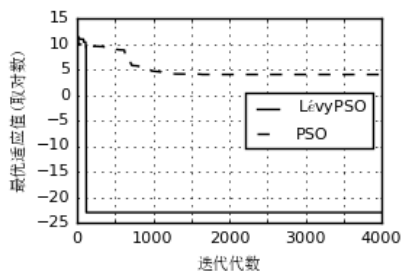


图 2 F2 函数进化曲线

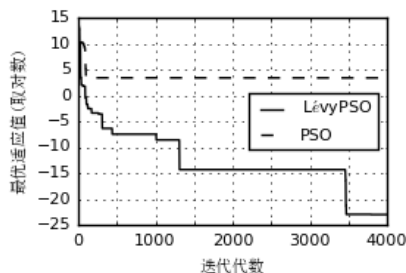


图 3 F3 函数进化曲线

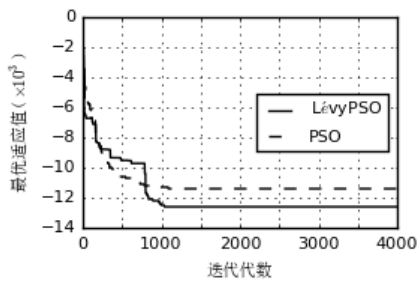


图4 F8 函数进化曲线

### 3.2 LévyPSO 与其他算法比较

为了验证 LévyPSO 算法的有效性, 将 LévyPSO 算法在 8 个函数上的优化结果与其他文献中的优化结果进行比较. 表 4 中其他各算法的数据从各参考文献中取得, 空白格说明算法没有在相应函数上的测试, 表中括号中的数据为相应实验的标准差. 比较结果显示, 除了在 F5 和 F6 函数上与 GPSO<sup>[27]</sup>持平, LévyPSO 都优于其他算法. 其中 AEPSo<sup>[18]</sup>结合动态调整的阈值和均匀变异, 因频繁变异且均匀变异不能利用粒子自身信息而降低了局部探索能力; LFPSO<sup>[22]</sup>和 GPSO+CJ<sup>[15]</sup>都是基于随机选择的粒子进行变异, 不能保证对“劣”粒子进行变异, 从而影响算法性能; GPSO 则是用灰关联分析对  $w$ 、 $c_1$ 、 $c_2$  进行自适应改变, 没有变异行为.

表 4 LévyPSO 与其他文献中的算法比较

函数	F1	F2	F3	F4
LévyPSO	<b>0</b> (0)	<b>0</b> (0)	<b>8.98E-06</b> (3.03E-05)	<b>1.64E-04</b> (1.52E-04)
AEPSo	2.00E-122 (8.80E-122)	1.20E-09 (1.40E-09)	14 (19.91)	
LFPSO			5.89 (89.9)	
GPSO			21.95 (1.46E-02)	2.46E-03 (2.67E-03)
函数	F5	F6	F7	F8
LévyPSO	<b>0</b> (0)	<b>0</b> (0)	<b>0</b> (0)	<b>-11483.6</b> (1091.87)
AEPSo	1.20E-02 (1.60E-02)	0.577 (1.084)	1.20E+02 (39.39)	
LFPSO	3.08E-02 (4.84E-02)	35.9 (15.5)		-7389.5 (758)
GPSO+CJ	0.0263 (0.03)	12.77 (4.864)		-10843 (295.3)
GPSO	<b>0</b>	<b>0</b>		-10477.75

### 3.3 不同变异算子的比较

为了比较不同变异算子的性能, 分别引入高斯变异 (GaussianPSO)、柯西变异 (CauchyPSO) 和均匀变异 (UniformPSO) 算子. 表 5 是 4 个不同变异算法的实验结果. 在 F1、F2、F5、F6 和 F7 五个函数上 LévyPSO、GaussianPSO 和 CauchyPSO 都能够完全收敛于全局极值; 对于 F3、F4 和 F8 函数, LévyPSO、GaussianPSO 和 CauchyPSO 的仿真结果基本一致; UniformPSO 除了在 F5 函数上与其他三种算法相当, 在另外 7 个函数上都逊于其他三种算法.

表 5 各随机变异算法实验结果

函数	Lévy	Gaussian	Cauchy	Uniform
F1	<b>0</b>	<b>0</b>	<b>0</b>	4.01E+04
	0	0	0	8.73E+03
F2	<b>0</b>	<b>0</b>	<b>0</b>	1.06E+03
	0	0	0	2.09E+03
F3	8.98E-06	7.90E-06	<b>6.25E-07</b>	9.38E-01
	3.03E-05	1.45E-05	2.48E-06	6.50
F4	1.64E-04	<b>1.26E-04</b>	1.44E-04	3.32E-04
	1.52E-04	1.08E-04	1.29E-04	3.65E-04
F5	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	0	0	0	0
F6	<b>0</b>	<b>0</b>	<b>0</b>	7.25
	0	0	0	14.3
F7	<b>0</b>	<b>0</b>	<b>0</b>	4.73E-01
	0	0	0	2.57
F8	-11483.6	-11478.4	<b>-11935.1</b>	-10651.3
	1091.87	1033.48	839.85	881.98

表 6 是算法的可靠性数据, 可靠性是指算法收敛于全局极值的次数占总实验次数的比例, 其中 F8 函数是指适应度值小于等于 -12569.48. 从数据看, 4 个带变异算子的 PSO 算法的可靠性明显高于 PSO 算法; LévyPSO、GaussianPSO 和 CauchyPSO 三个算法的可靠性基本相同并且高于 UniformPSO.

表 6 各算法可靠性的实验结果(%)

函数	PSO	Lévy	Gaussian	Cauchy	Uniform
F1	2	<b>100</b>	<b>100</b>	<b>100</b>	72
F2	0	<b>100</b>	<b>100</b>	<b>100</b>	80
F3	0	4	<b>6</b>	0	0
F4	0	0	0	0	0
F5	68	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
F6	6	<b>100</b>	<b>100</b>	<b>100</b>	72
F7	6	<b>100</b>	<b>100</b>	<b>100</b>	40
F8	0	44	36	<b>56</b>	6

综上所述, LévyPSO、GaussianPSO、CauchyPSO 比 UniformPSO 的性能更优, 不管是全局搜索能力还是可靠性.

### 3.4 多样性分析

LévyPSO 算法的人为机制和随机机制能够增加种群的多样性. 图 5 和图 6 描绘了 LévyPSO 和 PSO 分别在 F3 和 F8 函数上的位置、速率和方向上的多样性, 其中各多样性的测度参考<sup>[28]</sup>. 李国等指出一个高性能算法的多样性曲线拥有较大的频率和振幅, 尤其是进化前期, 而不是保持较大的多样性<sup>[29]</sup>. 图中显示 LévyPSO 在进化早期有更大的频率和振幅, 进化过程中能保持多样性的持续变化, 有更强的全局搜索能力; PSO 在进化早期的频率和振幅较小, 且位置多样性和速率多样性随着进化过程其频率和振幅迅速下降到接近 0, 虽然其方向多样性曲线的频率和振幅有所增强, 但已失去速率的粒子极易陷入局部极值而无法跳出.

### 3.5 LévyPSO 算法参数分析

#### 3.5.1 MSI 参数

表 3 中数据显示 LévyPSO 对 F3 和 F8 函数还没有达到最优性能. 由于 F3 函数在两边陡峭山峰的中间是一处平坦的山谷, 没有更多的搜索方向信息, 优化算法在该函数上很难获得它的最优值, 这时可考虑适当的增加最大允许停滞代数 MSI, 增加粒子在变异之前的搜索次数, 以确定最优的搜索方向. F8 函数局部极值点相距较远, 使得优化算法易陷入局部极值, 这时可考虑适当的减小最大允许停滞代数 MSI, 增加群体变异次数使得粒子尽快跳出局部极值. 表 7 显示不同 MSI 的取值时的实验结果, 最小值(Min)后括号中是搜索到该最小值的次数. 对于 F3 函数, MSI=20 时平均适应度值与 MSI=10 时相差不多, 但可靠性明显提高; 对 F8 函数则取 MSI=5.

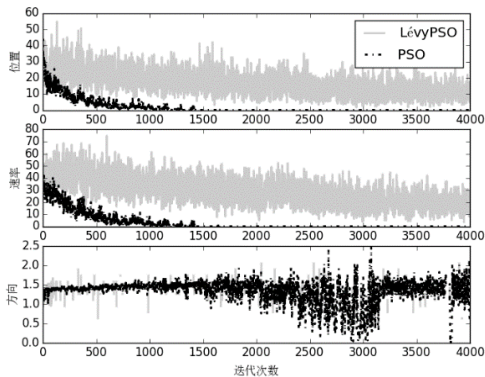


图 5 F3 函数上的多样性曲线

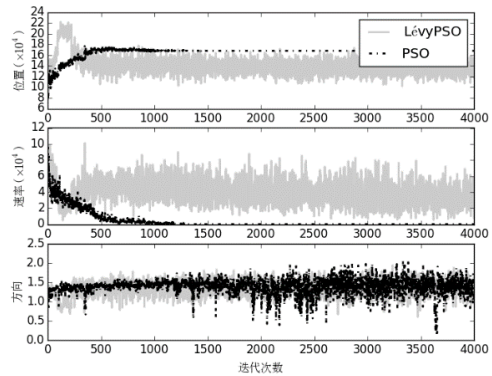


图 6 F8 函数上的多样性曲线

表 7 LévyPSO 算法中 MSI 参数设置比较

函数	MSI	Min	Mean	Std.Dev.
F3	5	1.89E-10	2.20E-05	4.68E-05
	10	1.94E-22	1.40E-06	5.33E-06
	20	0(16)	2.27E-06	7.49E-06
F8	5	0(10)	5.55E-05	0.000289
	10	-12569.48(35)	-12270.71	718.23
	20	-12569.48(24)	-11656.86	1026.44
F8	50	-12569.48(10)	-10970.40	1096.23
	50	-12331.09	-9814.21	1122.41

#### 3.5.2 进化代数

因 LévyPSO 能够保持良好的多样性, 所以可通过增加进化代数来提高搜索精度. 表 8 显示不同进化代数的搜索结果, 可以看出随着进化代数增加, 搜索精度随着提高, 但是运行时间也相应延长. 当进化代数为 1E+05 时, F8 函数都能收敛到全局最优解, F3 函数也有更小的均值和标准差.

表 8 LévyPSO 算法进化代数参数设置比较

函数	进化代数	Min	Mean	Std.Dev.
F3	4000	0(16)	2.27E-06	7.49E-06
	10000	0(25)	1.34E-06	5.80E-06
	50000	0(33)	5.55E-08	3.72E-07
F8	100000	0(30)	1.17E-10	5.05E-10
	4000	-12569.49(35)	-12270.71	718.23
	10000	-12569.49(38)	-12415.43	433.63
F8	50000	-12569.49(49)	-12567.12	16.76
	100000	-12569.49(50)	-12569.49	1.96E-06

#### 3.5.3 λ 参数

表 9 是变异尺度参数 λ 从 1 线性递减到 λ<sub>min</sub> 的实验结果, 对于单峰的 F3 函数, 在进化后段粒子都聚集

在全局最优点附件,这时变异的步伐要小才能有利于搜索到更好的解,表中也显示当 $\lambda_{\min}$ 等于0.1时寻优结果最好;而对于多峰函数F8来说,粒子易陷入局部极值点,如果离全局极值点较远,这时短距离的变异不能保证跳出局部极值点,表中也可以看出不同的 $\lambda_{\min}$ 参数的设置对算法的性能影响有限.因此,当优化问题是单模态函数时,变异尺度参数可采用线性递减的策略,可设 $\lambda$ 在[0,1]区间内递减;而当优化问题是多模态函数时,则可采用固定的变异尺度参数,即 $\lambda=1$ .

表9 LévyPSO算法 $\lambda$ 参数设置比较

函数	$\lambda_{\min}$	Min	Mean	Std.Dev.
F3	1	0(16)	2.27E-06	7.49E-06
	0.5	0(19)	4.71E-06	3.02E-05
	0.2	0(22)	1.87E-06	8.43E-06
	0.1	0(25)	2.56E-07	1.59E-06
F8	1	-12569.49(35)	-12270.71	718.23
	0.5	-12569.49(33)	-12173.70	732.24
	0.2	-12569.49(34)	-12275.52	674.49
	0.1	-12569.49(32)	-12142.92	829.30

#### 4 结论

本文针对 PSO 算法易早熟、易局部收敛的缺点,给出了一种新的基于 Lévy 变异的改进 PSO 算法—LévyPSO 算法. LévyPSO 算法定义了粒子的停滞状态并给出粒子变异的允许停滞代数准则,通过最大允许停滞代数来控制变异的频率,从而在探索与开发之间平衡;同时利用 Lévy 变异算子特征,在变异时充分利用粒子的自身信息,有效提高了算法的收敛精度和可靠性.仿真实验显示:①LévyPSO 算法的性能明显好于 PSO 算法;与其他的改进 PSO 算法相比,LévyPSO 算法都有相当或更好的表现;②基于高斯变异的 GaussianPSO 和柯西变异的 CauchyPSO 与 LévyPSO 有相同的性能,都好于基于均匀变异的 UniformPSO,说明了先在粒子附近搜索并偶尔长距离的跳转的算法思想是有效的;③通过 Lévy 变异算子能够保持 PSO 的多样性,保证全局收敛;④应用 LévyPSO 算法时,建议参数的设置:MSI=10, $\lambda=1$ (对复杂的单峰函数可采用线性递减的策略),Lévy 分布的参数可设 $\alpha=1.5$ 、 $\gamma=1$ (可直接使用高斯分布和柯西分布).在后续研究中应进一步分析最大允许停滞代数和

Lévy 变异对粒子行为的影响,研究不同选择准则对算法的影响.

#### 参考文献

- Kennedy J, Eberhart R. Particle swarm optimization. Proc. IEEE International Conference on Neural Networks, 1995, 4: 1942–1948.
- Shi Y, Eberhart R. A modified particle swarm optimizer. the 1998 IEEE International Conference on Evolutionary Computation. Proc. 1998 IEEE World Congress on Computational Intelligence. 1998, 69–73.
- Alrashidi MR, El-Hawary ME. A survey of particle swarm optimization applications in electric power systems. IEEE Trans. on Evolutionary Computation, 2009, 13(4): 913–918.
- Stimac G, Braut S, Zigulic R. Comparative analysis of PSO algorithms for PID controller tuning. Chinese Journal of Mechanical Engineering, 2014, 27(5): 928–936.
- Yusup N, Zain AM, Hashim SZM. Overview of PSO for optimizing process parameters of machining. Procedia Engineering, 2012, 29: 914–923.
- Kiranyaz S, Ince T, Gabbouj M. Adaptation, learning, and optimization: Multidimensional particle swarm optimization for machine learning and pattern recognition. Adaptation Learning & Optimization, 2013, 22(10): 1448–1462.
- Mahor A, Prasad V, Rangnekar S. Economic dispatch using particle swarm optimization: A review. Renewable and Sustainable Energy Reviews, 2009, 13(8): 2134–2141.
- 岑翼刚,孙德宝,李宁.WNN 中的改进 PSO 算法及参数初始化.华中科技大学学报(自然科学版),2006,34(8):43–45.
- Qu B, Liang J, Suganthan P. Niching particle swarm optimization with local search for multi-modal optimization. Information Sciences, 2012, 197: 131–143.
- Liu Y, Niu B, Luo Y. Hybrid learning particle swarm optimizer with genetic disturbance. Neurocomputing, 2015, 151: 1237–1247.
- 孙俊,方伟,吴小俊,等.量子行为粒子群优化:原理及其应用.北京:清华大学出版社,2011.
- Ling SH, Iu HH, Chan KY, et al. Hybrid particle swarm optimization with wavelet mutation and its industrial applications. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 2008, 38(3): 743–763.

- 13 Wan C, Wang J, Yang G, et al. Gaussian particle swarm optimization with differential evolution mutation. *Advances in Swarm Intelligence*. 2011. 439–446.
- 14 Higashi N, Iba H. Particle swarm optimization with Gaussian mutation. *Proc. of the 2003 Swarm Intelligence Symposium*, 2003. SIS'03. IEEE. 2003. 72–79.
- 15 Krohling RA. Gaussian particle swarm with jumps. *The 2005 IEEE Congress on Evolutionary Computation*, 2005. 2005, 2. 1226–1231.
- 16 Krohling RA, Santos D, Coelho L. PSO-E: Particle swarm with exponential distribution. *IEEE Congress on Evolutionary Computation*, 2006. 2006. 1428–1433.
- 17 Krohling RA, Mendel E. Bare bones particle swarm optimization with Gaussian or Cauchy jumps. *IEEE Congress on Evolutionary Computation*, 2009. 2009. 3285–3291.
- 18 赫然,王永吉,王青,等.一种改进的自适应逃逸微粒群算法及实验分析. *软件学报*,2005,16(12):2036–2044.
- 19 Li C, Liu Y, Zhou A, et al. A fast particle swarm optimization algorithm with cauchy mutation and natural selection strategy. *Advances in Computation and Intelligence*, 2007: 334–343.
- 20 Wang H, Liu Y, Li C, et al. A hybrid particle swarm algorithm with Cauchy mutation. *Swarm Intelligence Symposium*, 2007. 2007. 356–360.
- 21 Wang H, Liu Y, Wu Z, et al. An improved particle swarm optimization with adaptive jumps. *IEEE Congress on Evolutionary Computation*, 2008. 2008. 392–397.
- 22 Jana ND, Sil J. Particle swarm optimization with Lévy flight and adaptive polynomial mutation in gbest particle. *Advances in Intelligent Systems and Computing 235: Recent Advances in Intelligent Informatics*. Springer, 2014: 275–282.
- 23 Reynolds AM. Cooperative random Lévy flight searches and the flight patterns of honeybees. *Physics Letters A*, 2006, 354(5/6): 384–388.
- 24 Mantegna R. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Physics Review E*, 1994, 49(5): 4677–4683.
- 25 Solis FJ, Wets JB. Minimization by random search techniques. *Mathematics of Operations Research*, 1981, 6(1): 19–30.
- 26 Zhang Y, Gong D, Sun X, et al. Adaptive bare-bones particle swarm optimization algorithm and its convergence analysis. *Soft Computing*, 2014, 18(7): 1337–1352.
- 27 Leu MS, Yeh MF. Grey particle swarm optimization. *Applied Soft Computing*, 2012, 12(9): 2985–2996.
- 28 Shi Y, Eberhart R. Population diversity of particle swarms. *IEEE Congress on Evolutionary Computation*. 2008. 1063–1067.
- 29 李国,徐晨,吴延科.全局收敛的 PSO 算法的种群多样性特征. *计算机应用与软件*,2008,25(5):237–240,259.