

6-1-2020

Improved Particle Swarm Optimization Based on Lévy Flights

Rongyu Li

College of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China;

Wang Ying

College of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Improved Particle Swarm Optimization Based on Lévy Flights

Abstract

Abstract: The particle swarm optimization (PSO) has some demerits, such as relapsing into local extremum, slow convergence velocity and low convergence precision in the late evolutionary. The Lévy particle swarm optimization (Lévy PSO) was proposed. *In the particle position updating formula, Lévy PSO eliminated the impact of speed on the convergence rate, and used Levy flight to change the direction of particle positions movement to prevent particles getting into local optimum value, and then using greedy strategy to update the evaluation and choose the best solution to obtain the global optimum.* The experimental results show that Lévy PSO can effectively improve the accuracy and convergence speed and the Lévy PSO has better optimization effect than PSO, Cuckoo Search (CS) and Artificial Bee Colony Algorithm (ABC).

Keywords

particle swarm optimization, Lévy flights,, greedy strategy, optimization

Recommended Citation

Li Rongyu, Wang Ying. Improved Particle Swarm Optimization Based on Lévy Flights[J]. Journal of System Simulation, 2017, 29(8): 1685-1692.

基于莱维飞行的改进粒子群算法

李荣雨, 王颖

(南京工业大学计算机科学与技术学院, 江苏 南京 211816)

摘要: 针对基本粒子群算法容易发生早熟收敛, 陷入局部极值, 进化后期收敛速度慢以及精度低等缺点, 提出了基于莱维飞行的改进粒子群算法。在粒子位置更新公式中, 消除速度项对收敛速度的影响, 利用莱维飞行改变粒子位置移动方向, 防止粒子陷入局部最优值, 通过贪婪的更新评价策略, 选择最优解, 从而得到全局最优。实验结果表明, 与基本粒子群算法、布谷鸟搜索算法以及蜂群算法相比, 所提出的基于莱维飞行的改进粒子群算法能够有效地提高解的精度并加快收敛速度, 寻优效果更优。

关键词: 粒子群算法; 莱维飞行; 贪婪策略; 优化

中图分类号: TP301.6 文献标识码: A 文章编号: 1004-731X (2017) 08-1685-08

DOI: 10.16182/j.issn1004731x.joss.201708007

Improved Particle Swarm Optimization Based on Lévy Flights

Li Rongyu, Wang Ying

(College of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China)

Abstract: The particle swarm optimization (PSO) has some demerits, such as relapsing into local extremum, slow convergence velocity and low convergence precision in the late evolutionary. The Lévy particle swarm optimization (Lévy PSO) was proposed. In the particle position updating formula, Lévy PSO eliminated the impact of speed on the convergence rate, and used Lévy flight to change the direction of particle positions movement to prevent particles getting into local optimum value, and then using greedy strategy to update the evaluation and choose the best solution to obtain the global optimum. The experimental results show that Lévy PSO can effectively improve the accuracy and convergence speed and the Lévy PSO has better optimization effect than PSO, Cuckoo Search (CS) and Artificial Bee Colony Algorithm (ABC).

Keywords: particle swarm optimization; Lévy flights; greedy strategy; optimization

引言

粒子群算法(Particle Swarm Optimization, PSO)是由 Kennedy 等^[1]提出的仿生智能优化算法。PSO 算法通过模拟鸟群觅食行为有效地求解最优化问

题。PSO 算法结构简单, 收敛速度快, 目前已经被应用于多种工程优化问题, 如寻找路径最优^[2], 神经网络的训练^[3], 系统控制参数优化^[4], 信号方向估计^[5]等, 在多种领域都具有应用研究价值。

近年来, 很多学者为提高 PSO 算法的性能, 对它作了很多的改进。但这些改进算法^[6-8]大多着眼于 PSO 的参数选择或某个参数的动态修改策略, 难以克服 PSO 算法易陷入局部极小的固有弱点。曾毅等^[9]将模式搜索算法嵌入到 PSO 算法中,



收稿日期: 2015-09-16 修回日期: 2016-01-05;
基金项目: 江苏省高校自然科学基金(12KJB510007);
作者简介: 李荣雨(1963-), 男, 山东临清, 博士, 副教授, 研究方向为面向流程工业的机器学习、模拟优化与统计监控; 王颖(1990-), 女, 江苏丹阳, 硕士生, 研究方向为工业过程控制与优化。

<http://www.china-simulation.com>

• 1685 •

使得模式搜索算法的局部搜索能力与 PSO 算法的全局寻优能力能够优势互补。Angeline^[10]和 Higashi 等人^[11]分别将遗传算法中的选择操作和杂交操作引入到 PSO 算法中,有效提高了算法的收敛性。Xin 等人^[12]将 PSO 算法和差分进化算法(Differential Evolution, DE)相结合,通过设定一个学习统计周期对该周期内的两种算法的性能进行分析,并依据统计结果更新 PSO 算法和 DE 算法被选中执行的概率,提高了算法的全局优化能力。刘俊芳等人^[13]还提出了以 PSO 算法和人工蜂群算法(Artificial Bee Colony, ABC)为基础的混合算法,PSO 和 ABC 共享全局历史最优解,并分别指导 2 个子种群并行进化,缩小了搜索范围,克服了 PSO 算法易陷入局部最优的缺点。2014 年, Xia 等人^[14]在种群搜索过程中定期对每维决策变量空间进行划分,通过对历史进化信息的统计分析来决定重点搜索区域,实现搜索空间的逐步收缩,进而达到提高搜索效率和求解精度的目的。

本文在研究中发现基本 PSO 算法之所以容易发生早熟收敛,是因为粒子群算法的搜索策略约束了粒子在后期的邻域搜索能力,导致粒子种群多样性丧失,大量粒子分布在局部最优的附近不能跳出。而布谷鸟算法中的 Lévy flights 策略能有效提高解的多样性,因此将 Lévy flights 策略融入粒子群算法中可以解决粒子群算法搜索精度低的问题。在基本粒子群算法的进化过程中,粒子的速度项不会影响进化过程。为了避免因设定该参数而影响粒子的收敛速度和搜索精度,本文消除了 PSO 算法中的速度项对收敛速度的影响并结合 Lévy flights 策略,提出了一种基于莱维飞行的改进粒子群算法(Lévy Particle Swarm Optimization, Lévy PSO)。改进的粒子群算法引入了布谷鸟搜索算法中的 Lévy flights 理念,在粒子每次更新完位置后不直接计算目标函数值,而是使用莱维飞行进一步更新个体的位置,而后再采用贪婪算法计算目标函数值,这样不仅减少了控制参数而且有效地提高了算法的优化性能。在标准测试函数实

验中,验证了 Lévy PSO 算法优于 PSO 算法和 CS 算法的良好优化性能。

1 基本 PSO 算法

考虑优化一个含 N 个变量的连续函数,其搜索空间为 d 维连续空间,对其使用 PSO 算法进行优化。设函数优化问题为

$$\min f(X), X = [x_1, x_2, \dots, x_N] \quad (1)$$

$$\text{s.t. } x_i \in [a_i, b_i], i = 1, 2, \dots, N \quad (2)$$

式中: $f(X)$ 为目标函数; d 为自变量 x_i 的维数; $[a_i, b_i]$ 为 x_i 的搜索范围。

在 PSO 算法中,每个粒子对应一个待优化问题的潜在解。粒子 P_i 有自己的位置和速度,分别表示为 N 维矢量 $X_i = [x_{i1}, x_{i2}, \dots, x_{iN}]$ 和 $V_i = [v_{i1}, v_{i2}, \dots, v_{iN}]$ 。 X_i 对应的目标函数值称为粒子的适应度值 f_i ,算法通过其来度量粒子的优劣程度; V_i 决定粒子运动的方向和距离。粒子 P_i 始终记忆自身目前找到的最优解,称为个体极值 P_i ; 了整个群体目前找到的最优解,称为全局极值 P_g 并通过追踪两者来更新自己。

PSO 算法初始随机生成一群粒子,然后通过迭代求解。迭代终止条件为预先确定的最大迭代次数或者为对优化结果的精度要求。在第 k 次迭代,粒子 P_i 按照如下两式来更新自己的速度和位置:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (3)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (4)$$

式中: c_1 和 c_2 是学习因子,通常取 $c_1 = c_2$; r_1 和 r_2 是介于(0,1)之间的随机数; v_{id}^k 和 x_{id}^k 分别是粒子 P_i 在第 k 次迭代中第 d 维的速度和位置,两者均被限制在一定范围内; p_{id}^k 是粒子 P_i 在第 d 维的个体极值的位置; p_{gd}^k 是群体在第 d 维的全局极值的位置。

基本粒子群算法存在很多缺陷,如对环境的变化不敏感,常常会受到 P_i 和 P_g 的影响而很难收敛到全局最优。实际上,许多高维空间中的复杂多峰函数在全局最优点附近往往分布着一些局部最优点,这些局部最优将很容易吸引住粒子群,因此算法容易发生早熟收敛而陷入局部最优,算法

优化精度难以提高。

2 基于莱维飞行的改进 PSO 算法 (Lévy PSO)

为克服基本 PSO 算法在运算后期由于种群多样性的快速丧失而造成的算法“早熟收敛”问题, 对第 k 代种群中的粒子加入 Lévy 飞行。通过 Lévy 飞行进行进一步的判断后, 对已陷入“早熟收敛”的粒子进行邻域搜索, 从而获得全局最优。

2.1 Lévy 飞行行为

Lévy 飞行是由 Lévy 提出, Benoist-Madelbrot 进行描述的一种稳定分布。大体上说, Lévy 飞行是一种模拟自然界中动物觅食的一个随机游走过程。Lévy 飞行具有以下特点:

1) 独立同分布的随机变量之和与随机变量本身具有相同的分布, 即具有统计的自相似性和随机分形;

2) 具有幂律渐进性 $\sim |x|^{(-1-\alpha)}$, 也就是所说的服从“重尾”分布;

3) 具有广义中心极限定理。对随机变量和分布来说, Lévy 飞行是极限的一种, 具有邻域的吸引, 当系统的进化或实现结果由大量的随机数的和决定时, 就会出现吸引;

4) 具有无限均值和无限方差。

随后 Lévy 飞行被应用到优化问题和发现最优解的研究^[16]当中, 结果表明, Lévy 飞行具有优越的性能。

2.2 Lévy PSO 算法原理

PSO 算法的缺陷在于前期具有较快的收敛速度但在后期收敛速度变得缓慢, 并且收敛精度较低。这主要是当 PSO 处于进化停滞时, 粒子群中的粒子都会出现“聚集”现象, 粒子种群多样性逐渐丧失, 算法陷入局部极值。很多学者提出了许多改进方法, 但是, 这些策略主要用于调整式(3)、式(4)中的参数 ω , c_1 和 c_2 。这些改进策略都不同程度地提高了收敛速度和精度, 但都不是从 PSO 算法收

敛于局部极值的根本原因上采取改进策略。通过研究发现, Lévy 飞行能有效的使粒子通过随机游走产生新的解, 经历新的搜索路径和领域。从而增加了种群的多样性, 提高发现更优解的概率, 解决容易陷入局部极值的问题。

Lévy 飞行对粒子的调整是通过改变粒子位置来更新粒子状态。在粒子群更新公式中速度项 v 保留了上代粒子的移动方向, 使得粒子位置的更新方向受到上代粒子位置的影响较重而丧失多样性, 从而易于陷入局部最优。同时这也与 Lévy 飞行的随机游走理念相违背。为避免 PSO 算法更新公式中速度项 v 对 Lévy 飞行调整粒子位置的影响, 本文采用如下公式对粒子群更新公式进行改进。

$$x_{id}^{k+1} = \omega x_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (5)$$

虽然 Lévy 飞行能够使解摆脱局部收敛, 重新搜索解的空间, 但无法保证其寻找到的新解一定优于原来的解。所以本文采用基于贪婪的更新评价策略, 只有能够改善当前解的更新值才会被接受。基于贪婪的更新评价策略保障算法能够利用进化过程中每一代的最优解引导算法进行局部搜索, 从而获得更高质量的解并提高算法的收敛速度。令

$$P_{id} = \omega x_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (6)$$

$$P'_{id} = P_{id} + \alpha \oplus Levy(\beta) \quad (7)$$

因此 Lévy PSO 算法将式(3)、式(4)改进为

$$x_{id}^{k+1} = \begin{cases} P_{id} (fit(P_{id}) < fit(P'_{id})) \\ P'_{id} (fit(P'_{id}) < fit(P_{id})) \end{cases} \quad (8)$$

式中: $\alpha = \alpha_0 (x_{id} - p_{gd})$ 是步长信息, 用于控制随机搜索的范围; α_0 是常数 ($\alpha_0 = 0.01$); \oplus 是点乘积 (entry-wise multiplications); $levy(\beta) \sim |x|^{(-1-\beta)}$ ($0 < \beta \leq 2$); $fit(x)$ 为 x 的适应度值。

2.3 Lévy PSO 算法流程

算法流程如下:

Step 1 初始化 Lévy PSO 算法参数, 初始化粒子的位置。给定种群的规模数 N , 最大的迭代次数;

Step 2 计算各个粒子的目标函数值, 并存储

历史最优位置 p_{gd}^0 ;

Step 3 对种群中的每一个粒子 P_i 根据式(6)更新 P_i 的位置; 计算 P_i 的适应值 fit_i ;

Step 4 若 $fit(x_{id})$ 优于 P_{id} 的适应值, 则更新 p_{id} 为 P_i 的当前位置;

Step 5 若 $fit(x_{id})$ 优于 P_{gd} 的适应值, 则更新 p_{gd} 为 P_i 的当前位置;

Step 6 根据式(7)更新 P_i 的位置;

Step 7 根据式(8)采用贪婪的更新评价策略;

Step 8 若 $fit(p'_{id})$ 优于 P_{id} 的适应值, 则更新 P_{gd} 为 P_i 的当前位置;

Step 9 若 $fit(p'_{id})$ 优于 P_{gd} 的适应值, 则更新 P_{gd} 为 P_i 的当前位置;

Step 10 判断算法是否达到最大迭代次数, 若达到则转向 step6, 否则转向 step5;

Step 11 更新个体的历史最优位置 p_{id}^k 及全局最优位置 p_{gd}^k ;

Step 12 输出最优解 p_{gd}^k 的相关信息, 算法结束。

3 实验仿真与分析

3.1 实验参数设置

为验证文中提及的改进算法的性能, 本文将 Lévy PSO、PSO、CS 和 ABC 算法进行对比实验。

在仿真实验中选择了 5 个具有代表性的测试函数: Sphere 函数、Schaffer 函数、Rastrigin 函数、

Griewank 函数、Ackley 函数。Sphere 函数是比较简单的单峰函数, 比较容易搜索到最优值。Schaffer 是比较复杂的单峰函数。多模态函数 Rastrigin、Ackley, Griewank 是复杂的非线性全局优化函数, 这对于测试算法的全局搜索性非常好。表 1 列出了这些函数的维数、取值范围、理论最优解以及函数定义。

在实验中, PSO 算法、Lévy PSO 算法的粒子种群数 N 均设为 40; 惯性权重 ω 设为 0.729 8, 学习因子 c_1 和 c_2 均设为 1.496 2; β 设为 1.5; 所有测试函数的维数均为 30, 如表 1 所示; 所有测试函数的最大迭代次数设置为 1 000, 详细参数设置如表 2 所示。

ABC 算法是近几年比较热门的算法, 而 CS 算法是近一段时间新兴的群智能优化算法, 其中也使用了莱维飞行。为了更好的验证 Lévy PSO 算法在群智能算法中的优化性能, 特此在仿真实验中测试了蜂群算法^[17] (ABC)和布谷鸟搜索算法^[15] (CS)的优化性能。

ABC 算法的种群大小设置为 50; 其中引领蜂和跟随蜂的个数均为 25; 函数的维数为 30; Limit 值均设置为 750。详细参数设置如表 3 所示。

CS 算法的鸟巢的数量设置为 30; 发现蛋的概率设置为 0.25; 函数维数设为 30。详细参数设置如表 4 所示。

表 1 标准测试函数

Tab. 1 Standard test function

测试函数	维数	取值范围	理论最优值	函数定义
Sphere	30	[-100,100]	0	$f(x) = \sum_{i=1}^D x_i^2$
Schaffer	30	[-100,100]	0	$f(x) = 0.5 + \frac{\sin^2(\sqrt{\sum_{i=1}^D x_i^2}) - 0.5}{(1 + 0.001(\sum_{i=1}^D x_i^2))^2}$
Rastrigin	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
Griewank	30	[-600,600]	0	$f(x) = \frac{1}{4000} (\sum_{i=1}^D (x_i - 100)^2) - (\prod_{i=1}^D \frac{x_i - 100}{\sqrt{i}}) + 1$
Ackley	30	[-32,32]	0	$f(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right)$

表 2 PSO、Lévy PSO 参数设置
Tab. 2 PSO、Lévy PSO parameter settings

算法	粒子种群数	惯性权重	学习因子	函数维数	β 值	最大迭代次数
PSO	40	0.729 8	1.496 2	30	1.5	1 000
Lévy PSO	40	0.729 8	1.496 2	30	1.5	1 000

表 3 ABC 算法参数设置
Tab. 3 ABC parameter settings

算法	蜜蜂种群数(CS)	引领蜂个数	跟随蜂个数	函数维数	Limit 值
ABC	50	25	25	30	750

表 4 CS 算法参数设置
Tab. 4 CS parameter settings

算法	鸟巢数量	发现蛋概率值	函数维数
CS	30	0.7	30

针对每个测试函数各个算法均运行 20 次并求其最优值、最差值、平均值、标准差。最优值、最差值、平均值以及标准差的比较见表 5~9。为了方便对比不同算法之间收敛速率的差异,图 1~5 给出了 Lévy PSO 算法、PSO 算法、CS 算法和 ABC 算法在 5 种不同测试函数下的进化过程曲线。图中横坐标为迭代次数,纵坐标为目标函数值。

表 5 Sphere 函数仿真比较结果
Tab. 5 Sphere function simulation comparison results

算法	平均值	标准差	最优值	最差值
Lévy PSO	0	0	0	0
PSO	1.385 6	0.458 5	1.044 0	2.252 6
CS	4.147 7E-3	4.147 7E-3	4.147 7E-3	3.900 1E-3
ABC	4.147 7E-3	4.147 7E-3	4.147 7E-3	3.900 1E-3

表 6 Schaffer 函数仿真比较结果
Tab. 6 Schaffer function simulation comparison results

算法	平均值	标准差	最优值	最差值
Lévy PSO	0	0	0	0
PSO	0.385 6	0.086 6	0.272 7	0.549 7
CS	0.388 8	0.029 6	0.345 5	0.414 7
ABC	0.470 1	0.017 0	0.451 7	0.485 5

表 7 Rastrigin 函数仿真比较结果
Tab. 7 Rastrigin function simulation comparison results

算法	平均值	标准差	最优值	最差值
Lévy PSO	0	0	0	0
PSO	44.251 7	12.323 7	30.644 5	67.250 4
CS	94.580 5	4.084 9	91.389 2	101.267 1
ABC	1.692 2	0.512 4	1.006 2	2.152 8

表 8 Griewank 函数仿真比较结果
Tab. 8 Griewank function simulation comparison results

算法	平均值	标准差	最优值	最差值
Lévy PSO	0	0	0	0
PSO	20.108 5	4.130 4	14.122 5	24.434 5
CS	8.422 0E-2	7.020 1E-2	2.983 2E-2	2.202 5E-1
ABC	1.116 6E-2	1.910 2E-2	4.760 6E-7	4.643 6E-2

表 9 Ackley 函数仿真比较结果
Tab. 9 Ackley function simulation comparison results

算法	平均值	标准差	最优值	最差值
Lévy PSO	8.474 9E-16	9.251 4E-17	7.324 2E-16	9.861 2E-16
PSO	4.588 6	0.953 6	3.524 1	6.248 9
CS	1.081 6	0.808 9	0.297 6	2.375 2
ABC	4.617 7E-5	1.234 5E-5	3.285 6E-5	6.675 3E-5

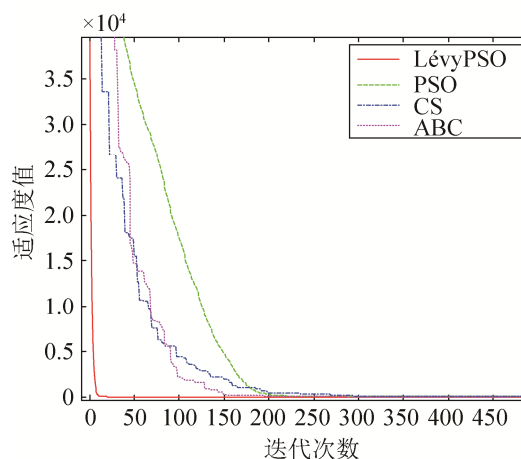


图 1 Sphere 函数进化曲线
Fig. 1 Sphere function evolution curve

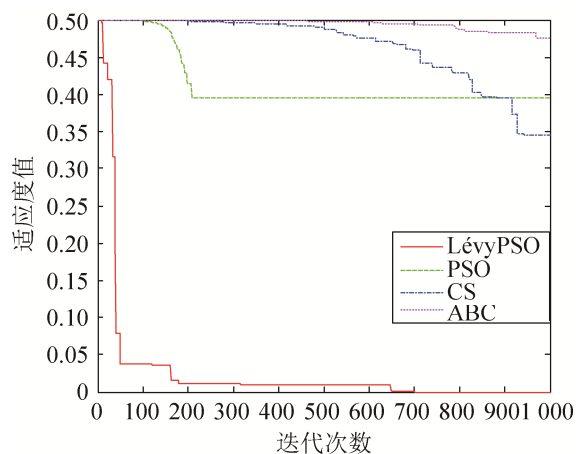


图 2 Schaffern 函数进化曲线
Fig. 2 Schaffern function evolution curve

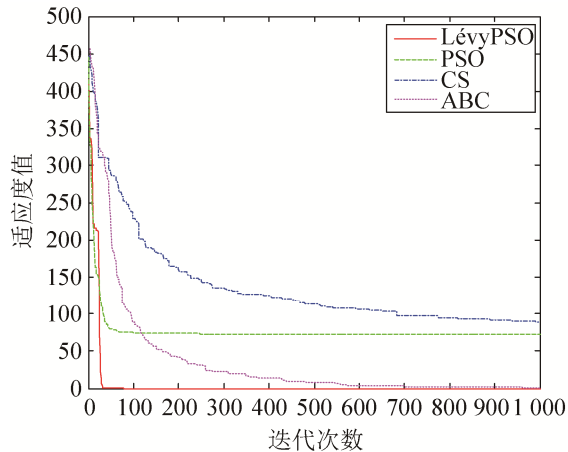


图 3 Rastrigin 函数进化曲线

Fig. 3 Rastrigin function evolution curve

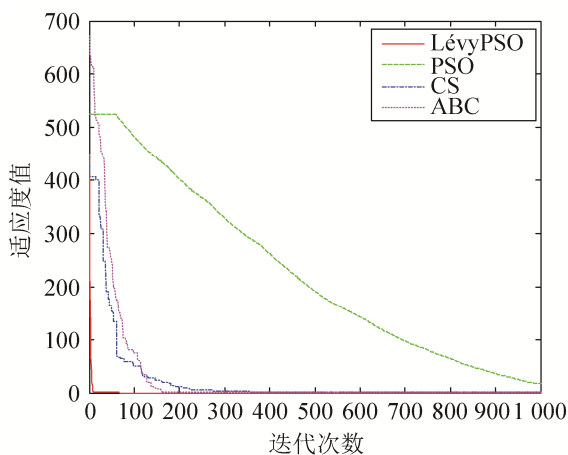


图 4 Griewank 函数进化曲线

Fig. 4 Griewank function evolution curve

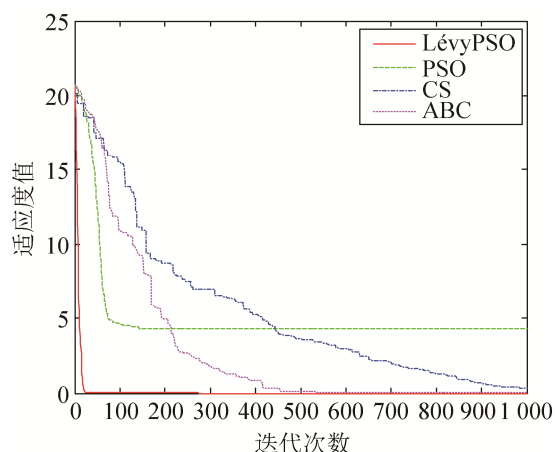


图 5 Ackley 函数进化曲线

Fig. 5 Ackley function evolution curve

3.2 实验数据分析

Sphere 函数是比较简单的单峰函数, 比较容易

搜索到最优值。由表 5 可知, 在对 Sphere 函数的仿真中 Lévy PSO 算法的解的精度最高, ABC 与 CS 算法次之, 而 PSO 算法精度最差。在算法的稳定方面, 明显 Lévy PSO 算法更为稳定。由图 1 可以看出 Lévy PSO 算法、ABC 算法和 CS 算法的收敛速度明显优于 PSO 算法, 而 Lévy PSO 算法的收敛速度也明显优于 ABC 算法和 CS 算法。

Schaffer 是比较复杂的单峰函数。由表 6 可知, 在对 Schaffer 函数的仿真中 Lévy PSO 算法的解的精度最高, PSO 与 CS 算法次之, 而 ABC 算法精度最差。在算法的稳定方面, 明显 Lévy PSO 算法更为稳定。从图 2 可知, Lévy PSO 算法的收敛速度明显优于另外三种种算法, 可以更快的找到最优值。

表 6~8 所反映的是关于多模态函数 Rastrigin、Ackley、Griewank 的优化问题仿真结果。以上函数均是复杂的非线性全局优化函数, 这对于测试算法的全局搜索性非常好。从表 7~9 可以看出 Lévy PSO 算法对与函数优化的精度均高于 PSO 算法、CS 算法和 ABC 算法, 特别是对于 Rastrigin 函数和 Ackley 函数的优化结果表明, PSO 算法、CS 算法和 ABC 算法均表现出了很差的优化性, 最终也并没有搜索到理论最优值, 而 Levy PSO 算法的搜索精度较 PSO 算法、CS 算法和 ABC 算法要精确很多。同时, 图 3~5 也表现出了 Lévy PSO 算法明显高于其他三种算法的搜索速度。所以对于 PSO 算法所存在的后期容易陷入局部最优的问题, Lévy PSO 算法已经有了很好的改进, 不仅如此, Lévy PSO 算法的收敛速度在 PSO 算法的基础上也得到了显著的提高。

上述所讨论的 Lévy PSO 与 PSO 算法的优化机理本质类似, 都是粒子从其个体及全局极值中获得更新信息, 并在此基础上进行随机或局部搜索。但是基于该机理的 Lévy PSO 算法实现局限于具体更新操作即位移更新算子, 该算子本身的局限性限制了算法在离散及组合优化领域的应用。针对连续参数优化问题的位移更新模型使用实数编码, 编码的每一维代表独立的变量, 不能反映编码中参数间的

顺序或其它约束关系。该模型的本质为粒子所代表的解在连续空间跟随其个体及邻域极值以矢量运算的形式进行更新。算法应用于整数规划问题时仅通过简单的截断法将更新后的连续解映射到离散解空间。此外, 算法的实数编码不适用于基于排序的组合优化问题, 且位移更新算子难以处理该类问题的顺序约束关系。

4 结论

本文在 PSO 算法的基础上, 提出一种 Lévy PSO 算法。该算法具有以下特点: 1) 引入 Lévy 飞行机制使算法能够在搜索后期摆脱局部收敛, 使得 PSO 算法在收敛速度快的同时能更好的提高收敛精度; 2) 采用贪婪的更新评价策略, 使得 PSO 算法能更好的提高收敛精度; 3) 通过仿真实验可以看出 Lévy PSO 算法相对 PSO 算法有更好的收敛精度和更快的收敛速度; 4) 无论是对于简单的单峰函数、复杂的单峰函数、多峰函数还是多模态函数, Lévy PSO 算法的搜索性能均高于 PSO 算法; 5) 和其他群智能算法的函数优化结果的比较也能看出以 Lévy PSO 算法的收敛精度和优化性能要远远优于 PSO 算法、CS 算法和 ABC 算法。

实验结果表明 Lévy PSO 算法具有较快的收敛速度和较高的收敛精度, 在不同的函数优化中都显现出优势, Lévy PSO 算法的优化性能已经高于 PSO 算法、CS 算法和 ABC 算法。

参考文献:

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [C]// Proceedings of the IEEE Conference on Neural Networks, IV. Perth, Australia: IEEE Press, 1995: 1942-1948.
- [2] Qin Y Q, Sun D B, Li M. Path planning for mobile robot using the particle swarm optimization with mutation operator [C]// Perth, Australia: Proc of Int Conf on Machine Learning and Cybernetics. 2004: 2473-2478.
- [3] Lu Jinna, Hu Hongping, Bai Yanping. Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and Ada Boost algorithm [J]. Neurocomputing (S0925-2312), 2015, 152: 305-315.
- [4] Panda S, Sahu B K, Mohanty P K. Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization [J]. Journal of the Franklin Institute-Engineering and Applied Mathematics (S0016-0032), 2012, 349(8): 2609-2625.
- [5] Meng Chaoli, Chen Shiawwu, Chang Ann-chen. Direction-of-Arrival Estimation Based on Particle Swarm Optimization Searching Approaches for CDMA Signals [J]. Wireless Personal Communications (S0929-6212), 2015, 81(1): 343-357.
- [6] 张顶学, 关治洪, 刘新芝. 一种动态改变惯性权重的自适应粒子群算法 [J]. 控制与决策, 2008, 23(11): 1254-1257. (Zhang D X, Guan Z H, Liu X Z. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight [J]. Control and Decision, 2008, 23(11): 1254-1257.)
- [7] 卢峰, 高立群. 基于改进粒子群算法的优化策略 [J]. 东北大学学报(自然科学版), 2011, 32(9): 1221-1224. (Lu F, Gao L Q. Novel optimization mechanism based on improved particle swarm optimization [J]. J. of Northeastern University(Natural Science edition), 2011, 32(9): 1221-1224.)
- [8] 姚灿中, 杨建梅. 基于变惯性权重及动态邻域的改进 PSO 算法 [J]. 计算机工程, 2011, 37(21): 20-22. (Yao C Z, Yang J M. Improved PSO algorithm based on variety inertia weight and dynamic neighborhood [J]. Computer Engineering, 2011, 37(21): 20-22.)
- [9] 曾毅, 朱旭生, 廖国勇. 一种基于领域空间的混合粒子群优化算法 [J]. 华东交通大学学报, 2013, 30(3): 44-49. (Zeng Yi, Zhu Xusheng, Liao Guoyong. Hybrid particle swarm optimization based on neighborhood space [J]. Journal of East China Jiaotong University, 2013, 30(3): 44-49.)
- [10] Angeline P J. Using selection to improve particle swarm optimization [C]// Proceedings of the IEEE International Conference on Evolutionary Computation, Alaska, USA. USA: IEEE, 1990: 84-89.
- [11] Higashi H, Iba H. Particle swarm optimization with Gaussian mutation [C]// Proceedings of the IEEE Swarm Intelligence Symposium, Indianapolis, USA. USA: IEEE, 2003: 72-79.
- [12] Xin B, Chen J, Peng Z H, et al. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization [J]. Science China (Information Sciences edition) (S1674-733X), 2010, 53(5): 980-989.