

应用随机游走和Boltzmann的烟花爆炸优化算法

李丽荣

河北地质大学艺术设计学院, 河北 石家庄

收稿日期: 2026年3月23日; 录用日期: 2026年4月21日; 发布日期: 2026年4月29日

摘要

为了克服烟花爆炸优化(Fireworks Explosion Optimization, FEO)算法容易早熟、解精度低的弱点, 提高算法的收敛速度, 提出了一种应用Lévy Flight机制的改进烟花爆炸优化(LFBFEO)算法。改进算法应用佳点集进行种群初始化, 并在爆炸变异算子中引入Lévy Flight机制产生新个体, 引导个体加速向最优个体靠近, 同时使个体具有摆脱局部极值约束的能力。应用Boltzmann机制实施子个体的生存选择, 以一定的概率接受劣质个体, 使算法避免早熟。最后, 在6组标准测试函数上的实验表明, LFBFEO克服了FEO算法的不足, 一定程度上提高了算法的收敛速度和收敛精度。

关键词

烟花爆炸优化, 早熟, 佳点集, Lévy Flight, Boltzmann

Improved Fireworks Explosion Optimizer Using Randomly Walk and Boltzmann

Lirong Li

School of Art Designing, Hebei GEO University, Shijiazhuang Hebei

Received: March 23, 2026; accepted: April 21, 2026; published: April 29, 2026

Abstract

To overcome the prematurity and low precision of Fireworks Explosion Optimizer (FEO) and improve its convergence speed, an improved FEO (LFBFEO) using Lévy Flight and Boltzmann mechanism was proposed. LFBFEO initializes the population by the good-point technique and uses Lévy Flight to improve the explosion operator of FEO, which not only can guide the individuals' approach

optimization fleetly, but also make the individuals have the ability to avoid the region of local optimization. Moreover, we introduce the Boltzmann mechanism into the selection operator. This mechanism makes some not-so-bad individuals accepted and makes the algorithm avoid premature. Finally, experiments were conducted on a set of 6 classical test functions. The experimental result demonstrates that the LBFEO improves the FEO and to some extent it improves the algorithm convergence speed and accuracy.

Keywords

Fireworks Explosion Optimizer, Prematurity, Good-Point Set, Lévy Flight, Boltzmann

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

自遗传算法(GA) [1]以来, 粒子群算法(Particle Swarm Optimization, PSO) [1]、差异演化(Differential Evolution, DE) [2]等智能优化算法的研究与应用, 为求解复杂 NP 问题等提供了有效、广泛的求解思路, 成为了人工智能领域的研究热点。优化算法亦可以用于实现对流言识别分类模型的参数寻优, 提升模型的分精度。

烟花搜索[3] (Fireworks search algorithm, FEA)算法模拟烟花爆炸的过程, 通过炸点在解空间中随机覆盖, 利用炸点所在位置信息的比较来实现寻优。在文献[3]中, 作者对其基本的搜索原理以及进化、变异机制进行了描述和分析。由于缺少严格的理论支持, 算法存在一些群智能算法所共有的弱点, 如: 早熟、求解精度低等。文献[4]针对 FEA 算法中的重要参数, 爆炸半径 r 进行了相关的改进, 以避免原公式中使 r 变为 0 的问题。通过对比, 改进算法较 FEA 具有更好的全局收敛能力。文献[5]针对 FEA 进行更加完善的改进和修正, 对爆炸半径、炸点产生方向等关键问题进行了详细的探讨和修改, 使其更具合理性和科学性, 提出了求解能力更强的烟花爆炸优化(Fireworks Explosion Optimization, FEO)算法, 并通过相关实验的对比, 验证了 FEO 算法的求解能力。文献[6]通过引入遗传算法的变异机制来改善 FEO 算法的收敛能力。迭代过程中, 随机选择其中某个炸点与当前最佳炸点位置进行信息交换, 并采用变异操作使执行变异操作的炸点与当前最优炸点共享进化信息。文献[7]针对算法对邻域搜索能力差的情况, 引入邻域搜索机制的改进 FEO 算法, 通过加入邻域搜索算子, 扩大解空间的搜索范围, 提高算法的解能力。文献[8]中, 作者借鉴粒子群算法中的个体交流思想, 在 FEO 算法的基础上借鉴了 PSO 算子中的交流机制, 通过该算子使每一代中每个炸点能够以一定的概率向目前已发现的最好的炸点位置移动, 以实现种群内个体之间优化信息的交流, 并在理论上证明了该算法的收敛性。

为了克服 FEO 的弱点, 提高其解精度, 本文提出一种应用 Lévy Flight [9]-[11]和 Boltzmann 的改进 FEO (Improved Fireworks Explosion Optimizer Using Lévy Flight and Boltzmann, LBFEO)算法。

2. 烟花爆炸优化算法

烟花爆炸过程中产生多个炸点覆盖解空间, 而且炸点位置具有随机性。源于此原理, 将待求解问题的解空间视作烟花爆炸的炸点覆盖问题, 爆炸进化过程中所产生炸点的位置信息视为问题的候选解, 优胜劣汰, 最终的进化结果将会使炸点集中于待优化问题的最优解附近, 最优解必然是问题的一个可选解。

烟花爆炸优化算法流程图如图 1 所示，公式(1)为爆炸变异算子产生新炸点。

$$E_i(t) = \{e_{i,j}(t) | e_{i,j}(t) = X_i(t) + r_j b_k\}, r_j \in [0, r]; \quad (1)$$

$b_k (k=1,2,\dots,m)$ 为方向向量。

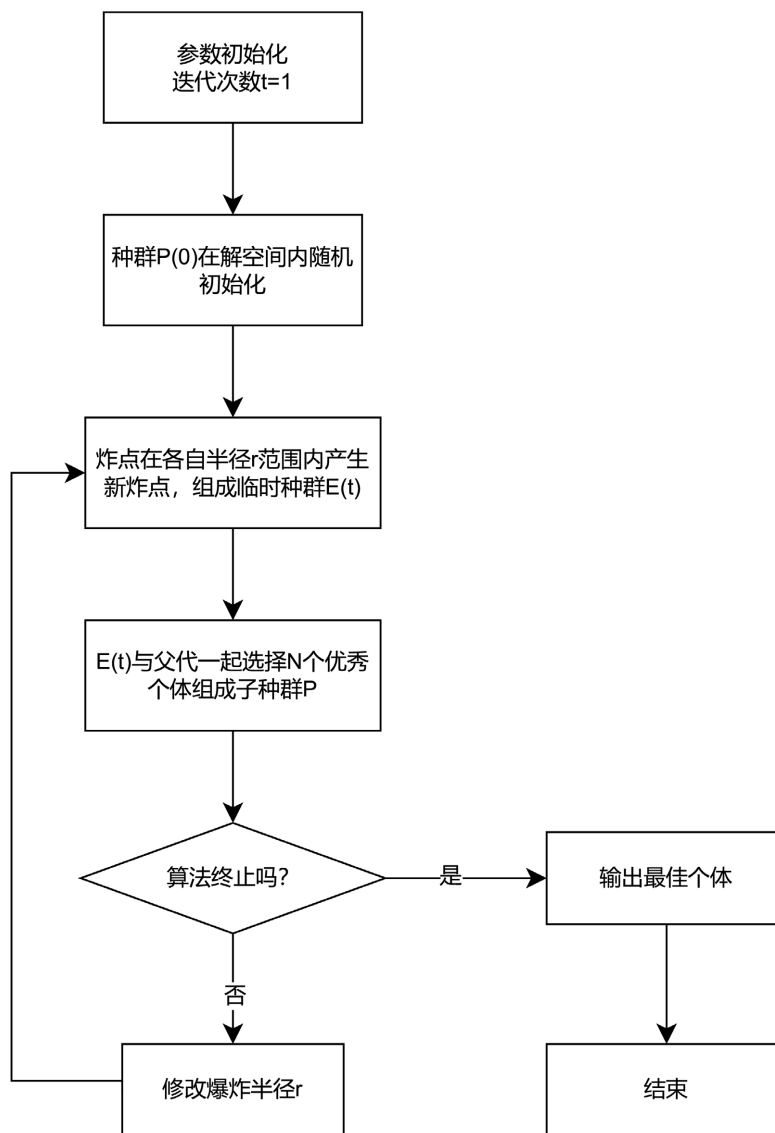


Figure 1. Flowchart of FEO

图 1. FEO 算法流程图

爆炸算子根据爆炸半径 r ，在多个方向上产生临时子炸点，文献[7]给出了产生炸点的基本方案。设解空间维度为 m ，当 $m \leq 5$ 时，父代 $X_i(t)$ 沿着标准坐标轴的 $2m$ 个方向，分别以半径 r 、 $2r/3$ 、 $r/3$ 产生临时子炸点。若 $m > 5$ 时，则从 m 个标准坐标轴随机挑选出互不相同的三组方向 X 、 Y 、 Z ，每组方向个数为 $m/5$ ，各方向上相应的反方向亦选择 $m/5$ 。

X 方向上按半径 r 产生子炸点，在 Y 方向上按 $2r/3$ 产生子炸点，在 Z 方向上按 $r/3$ 产生子炸点。爆炸半径 r 的调整见公式(2)。

$$r = \left(\frac{\text{iter}_{\max} - t}{\text{iter}_{\max}} \right)^k (r_0 - r_{\text{end}}) + r_{\text{end}} ; \quad (2)$$

算法的详细资料请参阅文献[7], 为了符合标准, 下面将使用“种群”、“个体”不再使用“炸点”等词语。

3. 应用 Lévy Flight 和 Boltzmann 的改进烟花爆炸优化算法

3.1. 基于佳点集初始化种群

考虑 FEO 算法采用爆炸变异机制生成子种群, 实现种群的进化, 爆炸半径 r 动态缩减。如果采用随机方式初始化种群, 由于种群在解空间内的分布呈现随机性, 难以保证通过爆炸算子所产生的炸点覆盖解空间, 初始爆炸半径 r_0 设定的科学性和合理性较难保证。如果在种群初始化时, 令种群在解空间内均匀分布, 假设待优化变量 $x \in [b, a]$, 则可令爆炸初始半径 $r_0 = (a - b) / 2$, 设置较为简单、合理, 而且解空间覆盖也比较广泛。根据文献[12]的研究结果可知, 利用佳点集[12][13]技术所产生的种群分布比随机初始化所产生的种群分布更为均匀, 种群的多样性更为丰富。所以, LFBFEO 采用佳点集进行种群初始化。

定义 1. 设 s 维欧式空间中存在某立方体 V_s , 存在形如

$P_n(k) = \left\{ \left\{ r_1^{(n)} * k \right\}, \left\{ r_2^{(n)} * k \right\}, \dots, \left\{ r_s^{(n)} * k \right\} \right\}, 1 \leq k \leq n$ 成立, 且其中 $r \in V_s$ 。如果其偏差 $\varphi(n) = C(r, \varepsilon) n^{-1+\varepsilon}$ 成立, 其中 ε 是任意正整数, $C(r, \varepsilon)$ 是只与 r, ε 有关的常数, 则称 r 为佳点, $P_n(k)$ 为佳点集。

一般情况下, 取 $r = \{2 \cos(2\pi k/p), 1 \leq k \leq s\}$, p 是 $(p-3)/2 \geq s$ 的最小素数, 则 r 是佳点。

3.2. Lévy Flight

Lévy Flight 是一种随机游走的运动机制, 研究成果表明, 果蝇、人类、光等均具有这种行为[9][10]。在该机制中, 行走物体能够将较长时间内的短距离搜索与偶尔的长距离搜索有机结合, 使两种搜索机制相间进行, 保证了对于解空间搜索的有效性和快速性。研究成果表明, 当目标位置随机, 并且稀疏时, 对于多个独立的搜索者, Lévy Flight 是最为理想的搜索策略之一[10]。Lévy 函数的概率分布函数为公式(3)表示。

$$P_{\alpha, \gamma}(x) = \frac{1}{\pi} \int_0^{\infty} \exp(\gamma q^\alpha) \times \cos(qx) dq \quad (3)$$

其概率密度函数可以通过其特征函数的连续傅里叶变换来定义, 见公式(4)。

$$g_{\alpha, \beta}(y) = \exp \left[iuy - \delta^\alpha |y|^\alpha \left(1 - i\beta \frac{y}{|y|} \tan \left(\frac{\pi}{2} \alpha \right) \right) \right] \quad (4)$$

公式(4)中 $\alpha \in (0, 2)$, 决定了该密度函数的分布特性, $\alpha \rightarrow 2$ 时, 随机变量服从正态分布, $\alpha = 1$ 时, 随机变量服从柯西分布, $\alpha = 0.5$, $\beta = 1$ 时, 服从 Lévy 分布。 $\beta \in [-1, 1]$, 当 $\beta > 0$ 时, Lévy 函数的概率分布对称, $\beta = 0$ 则概率分布偏右, $\beta < 0$ 时, 概率分布偏左。 u 为均值位移, δ 表示随机变量的发散程度。当 $0 < \alpha < 2$ 时, Lévy 分布的方差呈现发散并以指数形式增长, 其典型轨迹表现为较多的聚集短距离飞行和偶尔的长距离飞行相结合, Lévy 飞行过程中会发生大的跳跃且方向多变。

设 $X_{lbest}(t)$ 是第 t 次迭代过程中的最佳个体, 它包含了更多的引导种群向全局最优收敛的有益信息, 代表了种群的收敛方向。如果算法能够最终收敛于全局最优, 个体 $X_{lbest}(t)$ 所覆盖的搜索区域必然会收敛于全局最优所形成的搜索区域。另一方面, 考虑此时的当前最佳 X_{lbest} 如果是局部最优, 而该区域没有包含全局最优, 则算法可能会提前收敛。为了加速 FEO 算法的全局收敛, 一方面应该使其它个体在进化过

程中,向 \mathbf{X}_{lbest} 所在区域靠近,针对 \mathbf{X}_{lbest} 周围的邻域进行细微的局部搜索,另一方面应该在多次迭代过程中提供一定的进化变异机制,确保个体不被完全被约束在 \mathbf{X}_{lbest} 周围,能够及时搜索更大范围的空间。结合 FEO 算法的特点,将 FEO 算法的爆炸变异机制修改为公式(5)。

$$\mathbf{E}_i(t) = \mathbf{X}_i(t) + \text{Levy}(\lambda) \times (\mathbf{X}_i(t) - \mathbf{X}_{lbest}(t)) + r_j \mathbf{b}_k, r_j \in [0, r] \quad (5)$$

$\text{Levy}(\lambda)$ 表示跳跃步长服从 Lévy 分布的随机搜索向量。

Lévy Flight 机制的随机跳跃,有可能会使个体离开最佳 \mathbf{X}_{gbest} 较远。但考虑到 FEO 算法采用精英搜索策略,在进化过程中保留寻优过程中的历史最佳,所以经过较短时间调整后,远离最优解的个体会重新回到最优解附近。恰恰是因为存在着随机较大幅度的跳跃,才能够使种群中的个体具有较强的跳出局部最优约束的能力,同时还能够以较为精细的小步长勘探更好的解,在种群的多样性与集中性之间取得较好的平衡,算法性能得以提高。

3.3. 基于 Boltzmann 机制的子个体选择

原始 FEO 算法中采用典型的“爬山”策略生成子种群,即通过公式(5)产生子个体,并与父代一起择优生成子种群,这种贪心式进化机制,能够保证算法的收敛速度,也容易使算法陷入局部最优。实际应用中,往往存在适应度优的山峰周围遍布适应度劣的山峰的情况,种群到达峰顶必然会经过低谷,故劣质个体亦会包含有利于种群全局收敛和进化的优秀基因。给劣质个体生存机会,就会使有利于种群进化的优秀基因得以保留。

FEO 算法中父代个体依据公式(5)产生多个新子个体,这些子个体采取正向竞争的方式,选择适应度最优个体 $\mathbf{X}_i^b(t)$ 作为子种群的代表与父个体 $\mathbf{X}_i(t)$ 竞争。引入基于 Boltzmann 机制的选择操作,以一定的概率接受炸点 $\mathbf{X}_i^b(t)$,以求解最小值为例,概率计算公式(6)。

$$p = \left(\text{fit}_{\text{worst}}(t) - \text{fit}_i^b(t) \right) / \left(\text{fit}_{\text{worst}}(t) - \text{fit}_i(t) \right) \quad (6)$$

$\text{fit}_{\text{worst}}(t)$ 表示 t 代种群中最差子个体的适应度, $\text{fit}_i^b(t)$ 表示 $\mathbf{X}_i^b(t)$ 的适应度, $\text{fit}_i(t)$ 表示父个体 $\mathbf{X}_i(t)$ 的适应度。

当 $\mathbf{X}_i^b(t)$ 优于其父个体 $\mathbf{X}_i(t)$ 时, $p > 1$, 接受新个体 $\mathbf{X}_i^b(t)$, 并令其替换 $\mathbf{X}_i(t)$; 当 $\mathbf{X}_i^b(t)$ 劣于其父个体 $\mathbf{X}_i(t)$, 则以概率 p 接受 $\mathbf{X}_i^b(t)$, 并以其替换当前种群中的最差个体 $\mathbf{X}_{\text{worst}}(t)$ 。

3.4. LFBFEO 算法描述

算法 1. LFBFEO

不失一般性,以求解 $\min f(x_1, x_2, \dots, x_n)$, $x_i \in [L_i, U_i] (1 \leq i \leq n)$ 为例,个体位置状态表示为 $\mathbf{X} = (x_1, x_2, \dots, x_n)$, 将个体所处位置适应度函数设为 $f(\mathbf{X})$, 即待求解目标函数。

步骤 1. 初始化算法参数。最大迭代次数 $iter_{\text{max}}$, 种群规模 N , 初始爆炸半径 r_0 , 最终爆炸半径 r_{end} , 迭代次数 $t = 0$ 等;

步骤 2. 利用佳点集机制在解空间内生成全部初始种群 $\mathbf{P}(0) = \{\mathbf{X}_1(0), \mathbf{X}_2(0), \dots, \mathbf{X}_n(0)\}$;

步骤 3. 父个体 $\mathbf{X}_i(t)$ 按照公式(5)执行基于 LF 搜索的爆炸变异操作,生成种群 $E(t)$;

步骤 4. 从种群 $E(t)$ 内选择适应度最优的个体 $\mathbf{X}_i^b(t)$;

步骤 5. 依据公式(6)计算概率 $p = \left(\text{fit}_{\text{worst}}(t) - \text{fit}_i^b(t) \right) / \left(\text{fit}_{\text{worst}}(t) - \text{fit}_i(t) \right)$;

步骤 6. 如果 $p > 1$, 则令 $\mathbf{X}_i^b(t)$ 替换其父个体 $\mathbf{X}_i(t)$; 否则,以概率 p 接受 $\mathbf{X}_i^b(t)$, 替换掉种群内的最差个体 $\mathbf{X}_{\text{worst}}(t)$ 。

步骤 7. 父代个体执行爆炸变异操作没有结束,则 goto 步骤 3;

步骤 8. 修改第 $t+1$ 次迭代的爆炸半径 $r_{t+1} = \left(\frac{iter_{\max} - t}{iter_{\max}} \right)^k (r_0 - r_{end}) + r_{end}$;

步骤 9. 算法如果满足终止条件, 则输出当前最佳个体 $X_{gbest}(t)$, 并结束算法, 输出计算结果; 否则, goto 步骤 3 进入下一次迭代。

4. 仿真实验与分析

为了验证算法的有效性, 本文进行了大量的实验。个人计算机的配置为 AMD Turion 64 双核 CPU, CPU 主频为 2.0G, 计算机实际物理内存 1.87G。应用 Matlab6.5 平台对 LFBFEO 算法编程实现, 将其与 FEO 算法以及部分改进的烟花爆炸搜索算法(FEO [7]、GAFEO [8]、ESA [9]、PSOFEO [10])进行实验对比。所有参与实验对比算法的种群规模均设置为 30, LFBFEO 算法的爆炸半径参数设为 $r_0 = (a-b)/2$, $r_{end} = 10^{-6}$, 其它算法的参数设置参考各自相关文献。

为了公平, 所有算法的迭代次数均为 3000, 算法独立运行 30 次, 取 50 次运行结果的平均值进行比较。选择测试用的 6 个经典函数列于表 1。对比项目包括: 解的平均值(Mean)、解方差(std)、成功收敛的次数(num)、成功收敛的平均迭代次数(iter)。

Table 1. Benchmark functions

表 1. Benchmark 函数

编号	函数名称	函数表达式	取值范围
F1	Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	[-100, 100]
F2	Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]
F3	Schwefel	$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	[-100, 100]
F4	Rastrigin	$f_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]
F5	Ackely	$f_5(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{j=1}^n \cos(2\pi x_j) \right) + 20 + e$	[-32, 32]
F6	Griewank	$f_6(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1$	[-600, 600]

表 2 所列是参与对比的算法在函数维度为 30 时的求解结果。对于函数 f1 无论是解还是解的方差, LFBFEO、PSOFEO、GAFEO 三个算法均优于 FEO、ESA, 而在这三个算法中, LFBFEO 则是最高的。f2 函数是著名的陷阱函数, LFBFEO 与 PSOFEO 算法表现良好, 算法的解精度和解方差明显优于其它 3 个算法, LFBFEO 则是最好的。F3 函数用于检验算法的解精度, LFBFEO 稍逊于 PSOFEO, 但仍然是同一个精度级别, 要优于 GAFEO、FEO、GAFEO。在 f3 的解方差上, LFBFEO 则是最好的。在三个多峰

函数 f4、f5、f6, LFBFEO 算法无论是解精度还是解的方差都明显优于其它的相关算法。表 3 列出了在函数维度增加为 100 时, 5 个算法的求解精度和解方差。全部参与算法的求解能力较 30 维时均有所下降。LFBFEO 算法在 f1 上的解精度和解方差上较 PSOFEO 稍差, 但要远远优于其它算法。在其它的 5 个算法上, LFBFEO 算法表现出了较好的求解能力, 解精度和解方差都是最优的。表 4 列出了在 30、100 两种维度上, 参与对比算法的收敛成功的次数, 从所列数据可以看出, 在全部的函数上, 无论是低维还是高维, LFBFEO 的收敛次数都是最高的。

Table 2. Testing results on 30 dimension (Mean*std)
表 2. 30 维上的实验结果(Mean*std)

函数	FEO	GAFEO	ESA	PSOFEO	LFBFEO
F1	4.21e-20*3.71e-21	6.35e-58*7.21e-33	3.51e-36*4.16e-36	3.28e-59*7.32e-57	1.56e-59*3.81e-57
F2	2.19*8.62e-1	3.72e-3*9.01e-3	7.45e-5*1.24e-5	6.71e-6*7.36e-6	3.71e-6*6.37e-7
F3	8.17e-5*6.56e-4	5.13e-28*6.65e-26	3.21e-23*1.77e-20	4.42e-35*1.65e-35	7.78e-35*6.53e-36
F4	12.82e-2*0.72	4.12e-38*5.17e-36	5.76e-29*9.18e-30	1.89e-36*2.47e-38	1.78e-38*6.02e-40
F5	1.01e-3*0.33	1.89e-28*3.62e-24	6.73e-10*3.16e-11	7.72e-32*2.51e-26	5.27e-35*6.23e-25
F6	3.19e-2*6.53	8.10e-34*7.22e-34	6.32e-22*5.30e-20	1.10e-36*3.26e-35	9.01e-40*4.71e-41

Table 3. Testing results on 100 dimension (Mean*std)
表 3. 100 维上的实验结果(Mean*std)

函数	FEO	GAFEO	ESA	PSOFEO	LFBFEO
F1	3.82e-10*5.11e-9	6.12e-18*1.27e-17	7.10e-19*4.81e-15	7.16e-35*2.46e-31	3.54e-34*9.01e-31
F2	60.1*21.2	11.47*9.23	6.37*2.31	0.229e*0.654	8.01e-2*4.12e-3
F3	3.91*10.26	5.74e-12*4.52e-11	4.63e-10*1.91e-16	9.01e-15*2.18e-13	2.17e-16*9.45e-12
F4	7.67e-1*1.68	3.13e-21*5.14e-19	5.96e-13*9.04e-15	5.79e-30*1.76e-29	1.79e-32*9.02e-32
F5	19.89*11.23	1.89e-1*7.16e-3	3.71*16.11	5.27e-5*6.12e-4	7.15e-9*4.01e-11
F6	29.13*22.05	2.32e-3*5.63e-4	3.62*1.39	6.81e-6*9.18e-5	4.91e-11*2.93e-12

Table 4. Comparison of the number of successful convergences (num³⁰*num¹⁰⁰)
表 4. 算法收敛成功次数的对比(num³⁰*num¹⁰⁰)

Func	FEO	GAFEO	ESA	PSOFEO	LFBFEO
F1	15*10	28*19	25*16	28*22	28*22
F2	8*0	22*8	22*7	25*20	25*20
F3	12*0	23*12	22*13	26*21	26*22
F4	6*2	26*16	26*16	28*26	29*26
F5	8*0	14*12	9*7	26*12	28*17
F6	9*0	25*12	10*9	27*18	28*20

为了更直观地展示 LFBFEO 算法的收敛能力, 绘制了 FEO、LFBFEO 两个算法在 f1、f4、f5、f6 函数(30 维)上的收敛曲线。分别为图 2~5。从 4 个图的收敛曲线可以看出, LFBFEO 算法收敛曲线平滑, 随着迭代次数的增加, 下降迅速, 并最终收敛。从 4 个图形的 FEO 收敛曲线可以看出, FEO 的收敛曲线不如 LFBFEO 平滑, 甚至在于 f4、f5 两个函数上还出现有跳跃。

LFBFEO 算法针对爆炸变异算子引入 Lévy Flight 机制, 一方面能够加速种群向最佳个体收敛, 针对最优个体的周围进行搜索, 一方面个体具有摆脱局部最优束缚的能力, 扩大搜索范围。基于 Boltzmann 的选择算子, 赋予了劣质个体一定的生存机会, 维持了种群的多样性。综合以上多方面对比, 可知 LFBFEO 算法针对 FEO 的改进是成功的。

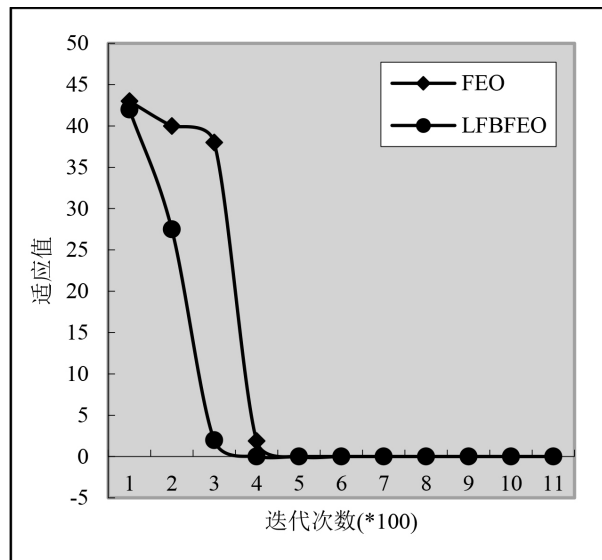


Figure 2. Convergence curve on f1

图 2. f1 函数上的收敛曲线

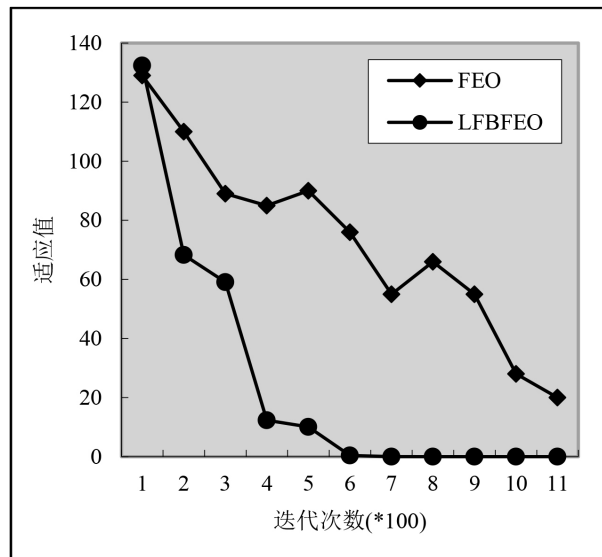


Figure 3. Convergence curve on f4

图 3. f4 函数上的收敛曲线

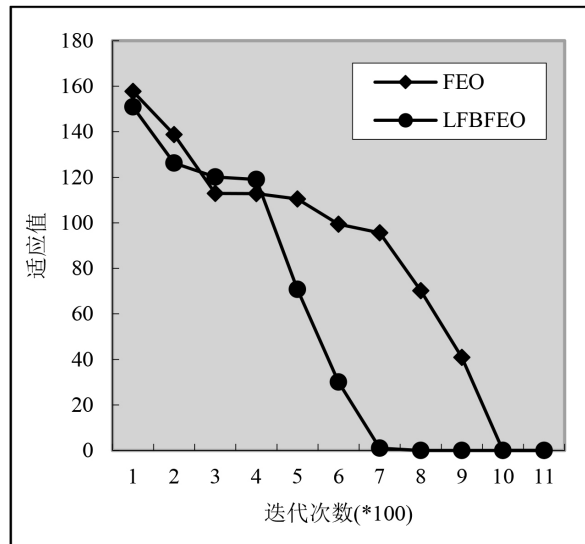


Figure 4. Convergence curve on f5
图 4. f5 函数上的收敛曲线

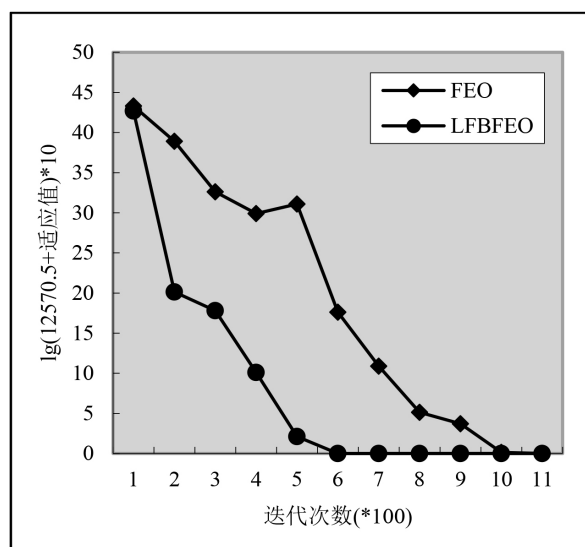


Figure 5. Convergence curve on f6
图 5. f6 函数上的收敛曲线

5. 结束语

引入佳点集、Lévy Flight、Boltzmann 三种机制分别对 FEO 的种群初始化、变异算子以及选择算子进行了改进，以克服算法容易早熟、解精度较低的弱点，提高算法的求解能力。通过一系列的测试表明，改进算法无论是求解质量还是收敛速度都具有一定的优势。FEO 算法出现时间较短，标准 FEO 算法的进化机制适合于连续性质的函数优化问题，进一步将针对如何求解离散性质的组合优化问题以及约束优化问题进行研究。

基金项目

河北省社会科学基金资助项目(HB21GL050)，《数据科学视域下公共卫生事件中流言识别及控制策

略研究》。

参考文献

- [1] Abdelkader, R.F. (2011) An Improved Discrete PSO with GA Operators for Efficient QoS-Multicast Routing. *International Journal of Hybrid Information Technology*, **4**, 23-38.
- [2] Sivanaithaperumal, S., Amali, S.M.J., Baskar, S. and Suganthan, P.N. (2011) Constrained Self-Adaptive Differential Evolution Based Design of Robust Optimal Fixed Structure Controller. *Engineering Applications of Artificial Intelligence*, **24**, 1084-1093. <https://doi.org/10.1016/j.engappai.2011.05.003>
- [3] Tan, Y. and Zhu, Y. (2010) Fireworks Algorithm for Optimization. In: Tan, Y., Shi, Y. and Tan, K.C., Eds., *Advances in Swarm Intelligence*, Springer, 355-364. https://doi.org/10.1007/978-3-642-13495-1_44
- [4] 杜振鑫. 烟花算法中爆炸半径的改进研究[J]. 计算机时代, 2013, 10(1): 28-29.
- [5] 曹炬, 贾红, 李婷婷. 烟花爆炸优化算法[J]. 计算机工程与科学, 2011, 33(1): 138-142.
- [6] 曹炬, 李婷婷, 贾红. 带有遗传算子的烟花爆炸优化算法[J]. 计算机工程, 2010, 36(23): 149-151, 154.
- [7] 曹炬, 侯学卿. 带有邻域搜索机制的爆炸搜索算法[J]. 计算机工程, 2011, 37(18): 183-184, 187.
- [8] 曹炬, 季艳芳. 改进的烟火爆炸优化算法及其收敛性分析[J]. 计算机工程与科学, 2012, 34(1): 90-93.
- [9] 刘长平, 叶春明. 具有 Lévy 飞行特征的蝙蝠算法[J]. 智能系统学报, 2013, 8(3): 240-246.
- [10] 谢健, 周永权, 陈欢. 一种基于 Lévy 飞行轨迹的蝙蝠算法[J]. 模式识别与人工智能, 2013, 26(9): 829-837.
- [11] Mantegna, R.N. (1994) Fast, Accurate Algorithm for Numerical Simulation of Lévy Stable Stochastic Processes. *Physical Review E*, **49**, 4677-4683. <https://doi.org/10.1103/physreve.49.4677>
- [12] 华罗庚, 王元. 数论在近似分析中的应用[M]. 北京: 科学出版社, 1978: 1-99.
- [13] 陈义雄, 梁昔明, 黄亚飞. 基于佳点集构造的改进量子粒子群优化算法[J]. 中南大学学报(自然科学版), 2013, 44(4): 1409-1414.