

# 融合多种策略的改进粒子群算法<sup>①</sup>

胡 佳

(中国市政工程中南设计研究总院有限公司, 武汉 430010)

通讯作者: 胡 佳, E-mail: 206092@whut.edu.cn



**摘 要:** 为有效解决粒子群优化算法 (Particle Swarm Optimization, PSO) 容易陷入局部极值及进化后期收敛速度慢、精度低等缺点, 提出了一种融合多种策略的改进粒子群算法 (Improved Particle Swarm Optimization, IPSO). 该算法包括以下 4 点改进: (1) 采取分组控制策略, 按适应度值将种群分为优解组和劣解组, 优解组进行遗传交叉操作, 劣解组进行变异操作; (2) 精英策略用来更新种群, 根据适应度值从经过交叉和变异操作后的种群及初始种群中选出前一半粒子作为新种群; (3) 改进粒子学习模式, 充分利用种群信息, 以优良种群的均值代替个体最优位置; (4) 引入概率控制来控制算法进入交叉和变异操作的概率. 测试函数的仿真结果表明, 与标准 PSO 及其改进算法相比, IPSO 算法能有效兼顾全局探索和局部挖掘能力, 具有收敛速度快、求解精度高、避开局部最优解的优点.

**关键词:** 粒子群算法; 分组策略; 精英策略; 概率控制

引用格式: 胡佳. 融合多种策略的改进粒子群算法. 计算机系统应用, 2021, 30(7): 172-177. <http://www.c-s-a.org.cn/1003-3254/7989.html>

## Improved Particle Swarm Optimization Integrating Multiple Strategies

HU Jia

(Central & Southern China Municipal Engineering Design and Research Institute Co. Ltd., Wuhan 430010, China)

**Abstract:** Particle Swarm Optimization (PSO) can easily fall into the local extremum and has slow convergence and low precision in the late evolution. For these reasons, we propose an Improved Particle Swarm Optimization (IPSO) algorithm that integrates multiple strategies. It includes the following four improvements. Firstly, the grouping strategy is adopted. According to the fitness values, the population is divided into an optimal particle group and an inferior particle group, which are subject to crossover and mutation operations, respectively. Secondly, the elite strategy is used to update the population. The first 50% particles are selected from the population after crossover and mutation operations and the initial population according to fitness values and taken as a new population. Thirdly, the particle learning mode is improved to make full use of the population information. The particle best is replaced with the mean of the optimal particle group. Fourthly, probability control is introduced to control the probability of the algorithm's entering crossover and mutation operations. The simulation results of the test functions show that compared with the standard PSO and its improved variants, the IPSO algorithm can effectively take into account the global exploration and local mining capabilities, and has the advantages of fast convergence, high accuracy, and avoidance from the local optimal solution.

**Key words:** Particle Swarm Optimization (PSO); grouping strategy; elite strategy; probability control

粒子群优化算法 (Particle Swarm Optimization, PSO) 是由 Kennedy 和 Eberhart<sup>[1]</sup> 提出的模拟鸟群觅

食行为的一种群体协作随机优化方法. PSO 算法中每个粒子都代表一个问题的候选解, 通过粒子个体的简

① 收稿时间: 2020-10-21; 修改时间: 2020-11-18, 2020-12-12; 采用时间: 2020-12-18; csa 在线出版时间: 2021-06-30

单行为及群体内的信息交互实现问题求解的智能性。其概念简单、实现容易、鲁棒性强、优化性能好,常用于解决多峰值、非线性等问题,并在函数优化<sup>[2]</sup>、图像处理<sup>[3]</sup>、系统辨识<sup>[4]</sup>、神经网络<sup>[5]</sup>、有限元模型修正<sup>[6]</sup>等工程和科学领域得到广泛应用。

虽然 PSO 算法优化性能较好,但也存在早熟收敛、容易陷入局部极值、迭代后期收敛速度慢等问题。对此,国内外研究学者从调整算法自身参数、改变粒子学习模式、结合其他优化算法等方向提出诸多改进的方法。吴静等人<sup>[7]</sup>通过动态调整惯性权重,提高了算法收敛速度及稳定性。张继荣等人<sup>[8]</sup>提出一种惯性权重余弦调整的 PSO 算法,提高了算法的精度。李龙澍等人<sup>[9]</sup>为平衡粒子的搜索行为提出了一种新的自适应惯性权重混沌 PSO 算法,提高算法全局探索能力。谭阳等人<sup>[10]</sup>改变粒子学习模式,将 PSO 中指导粒子群进化的全局最优解扩展为由多个精英粒子组成的精英子群,提高了算法跳出局部极值的能力。Liang 等人<sup>[11]</sup>提出了一种综合学习 PSO 算法,在多峰函数上具有较好的优化效果。Jiang 等人<sup>[12]</sup>将遗传算法中杂交算子引入到 PSO 算法中,有效提高了算法的收敛性。Xia 等人<sup>[13]</sup>在 PSO 算法中引入变异算子对人行桥进行模型修正,算法收敛速度和精度得到提高,得到令人满意的修正结果。刘波等人<sup>[14]</sup>提出了一种混合模拟退火算法的改进 PSO 算法,提高了算法收敛到全局最优解的速度和精度。张庭场等人<sup>[15]</sup>提出一种融合裂变和变异操作的分合群 PSO 算法,提高了算法局部收敛能力。这些改进算法大都要么提高了算法的收敛速度和精度,即提高算法局部挖掘能力,要么增强算法跳出局部极值的能力,即提高算法全局探索能力,很难同时兼顾两者。基于以上研究,为有效地兼顾了全局探索与局部挖掘能力,提出了一种融合多种策略的改进粒子群算法 (Improved Particle Swarm Optimization Algorithm, IPSO)。实验结果表明,IPSO 同其他 PSO 算法相比,收敛速度快、求解精度高且容易跳出局部最优解。

## 1 基本 PSO 算法

PSO 算法是一种基于群智能的随机搜索算法,源于对鸟群捕食行为的研究。PSO 算法由一群粒子组成,每个粒子可视为 D 维搜索空间中的一个搜索个体,具有自身的位置和速度,并且能够保留迄今为止所找到的最优位置以及所有粒子目前找到的全局最优位置。算

法迭代过程中,每个粒子根据个体最优位置和全局最优位置来更新自己的速度和位置,通过目标函数适应度值来评价粒子更新的位置是否更优,从而不断进化来获得新的个体最优和全局最优位置。在标准 PSO 算法中,粒子迭代更新公式如下:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{p}_g^t - \mathbf{x}_i^t) \quad (1)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (2)$$

式中,  $\mathbf{x}_i^t$  表示第  $t$  次迭代第  $i$  个粒子的位置,  $\mathbf{v}_i^t$  表示第  $t$  次迭代第  $i$  个粒子的速度,  $\mathbf{v}_i^{t+1}$  表示第  $t+1$  次迭代粒子的速度,  $\mathbf{x}_i^{t+1}$  表示第  $t+1$  次迭代粒子的位置,  $\mathbf{p}_i^t$  表示第  $i$  个粒子搜索至第  $t$  代时历史最优位置,即个体最优,  $\mathbf{p}_g^t$  表示搜索至第  $t$  代时整个种群历史最优位置,即全局最优,  $\omega$  是惯性权重,决定前一时刻的速度对下次移动的影响,  $c_1$ 、 $c_2$  为学习因子,通常取 2,  $r_1$ 、 $r_2$  为  $[0,1]$  内随机数,代表扰动因子。

惯性权重  $\omega$  对平衡算法的收敛速度和全局搜索能力有着重要的作用。  $\omega$  取值大有利于全局探索,并增加种群的多样性,而较小的  $\omega$  则可以提高算法的局部挖掘能力,加快收敛速度。为提高算法的前期全局探索能力和后期局部挖掘能力,本文采用常用的线性递减权重策略<sup>[2]</sup>,如式 (3) 所示:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{G_{\max}} \cdot t \quad (3)$$

其中,  $t$  为当前迭代次数,  $G_{\max}$  为最大迭代次数,  $\omega_{\max}$  为初始惯性权重,通常取 0.9,  $\omega_{\min}$  为最大迭代次数时的惯性权重,通常取 0.4。

粒子的飞行速度  $\mathbf{v}$  影响粒子位置更新的步长。最大飞行速度  $\mathbf{v}_{\max}$  一般取变量可行域的 10%~20%。迭代过程中,粒子的飞行速度将被约束在  $[-\mathbf{v}_{\max}, \mathbf{v}_{\max}]$  的范围内。

## 2 改进粒子群算法

本文融合了多种策略对基本 PSO 进行改进。首先,受达尔文“优胜劣汰、适者生存”生物进化思想<sup>[16]</sup>的启发,对种群粒子进行选择,适应度值小的优良种群被保留,其它粒子则被淘汰,具体实施中采取了分组策略和精英策略。按适应度值从小到大排序将种群分为优解组和劣解组,优解组进行遗传交叉操作,劣解组进行变异操作;然后将经过交叉变异后的种群和当前迭代次数下的初始种群按适应度值从小到大进行排序,选出前一半粒子作为更新种群。其次,采取改进粒子学习

模式的策略,充分利用了群体信息,保证了种群的多样性,提高了算法全局探索的能力.最后,采取了控制概率的策略,通过引入决策数  $P$ ,控制进入交叉变异等操作的概率,使算法在迭代初期小概率执行交叉变异等操作,尽量增加种群多样性,保证算法的全局探索的能力;随迭代次数的增加,进入交叉变异等操作的概率逐渐增大,迭代后期的收敛速度显著加快,算法的局部挖掘能力得到增强.通过多种策略的融合,IPSO 算法能有效兼顾全局探索和局部挖掘能力,具有收敛速度快、求解精度高、避开局部最优解的优点.

### 2.1 分组策略

根据目标函数的适应度值对种群进行分组.首先生成包含  $N$  个粒子的初始种群,计算种群内所有粒子目标函数的适应度值  $fit(\mathbf{x}_i)$  并排序,根据排序结果选择适应度值小的前一半粒子作为优解组,其他粒子作为劣解组,优解组粒子进行交叉操作,产生新的优良子代种群,如式 (4) 所示:

$$\hat{x}'_i = \eta \hat{x}_{index1(i)} + (1 - \eta) \hat{x}_{index2(i)}, i = 1, 2, \dots, \hat{N} \quad (4)$$

其中  $\hat{x}'_i$  表示产生新的优良子代种群,  $\hat{x}_i$  表示优解组中按适应度值排序的第  $i$  个粒子,  $\eta$  表示 (0,1) 间随机数,  $index1$ 、 $index2$  均表示无重复的随机排列数组,范围为从 1 到  $\hat{N}$  的整数,  $\hat{N}$  表示优解组种群粒子数,其大小为  $N/2$ .

劣解组进行变异操作,通过变异生成优良基因,充分利用每个粒子信息,如式 (5) 所示:

$$\tilde{x}''_i = \begin{cases} \tilde{x}_i + (\tilde{x}_i - x_{\max}) \times f(t), & r \geq 0.5 \\ \tilde{x}_i + (\tilde{x}_i - x_{\min}) \times f(t), & r < 0.5 \end{cases} \quad (5)$$

式中,  $\tilde{x}''_i$  表示从劣解组中产生的新种群,  $\tilde{x}_i$  表示劣解组中按适应度值排序的第  $i$  个粒子,  $f(t) = r_3(1 - t/G_{\max})^2$ ,  $x_{\max}$ 、 $x_{\min}$  分别表示粒子上下界,  $r_3$ 、 $r$  均表示 (0,1) 间随机数;  $t$  为当前迭代次数,  $G_{\max}$  为设置的迭代总次数.

### 2.2 精英策略

将优解组通过交叉生成的新种群  $\hat{x}'_i$  和劣解组通过变异生成的新种群  $\tilde{x}''_i$  合并,组成一个大小与初始种群相同的新种群  $\hat{x}'''_i$ ,再同当前迭代次数下的初始种群一起形成扩大种群,按适应度值排序,选出前一半新的优良粒子用于更新迭代.

### 2.3 改进粒子学习模式

个体除总结自身的实践经验和向最优个体学习外,也学习其他优秀同伴的行为,尤其在进化初期,这种行为在个体的学习中应处于主导地位.为充分利用群体

信息,从排序后的种群中选出前 1/4 粒子,以这些粒子的均值代替个体最优指导粒子搜索,以保证种群的多样性.对基本 PSO 算法中的个体极值  $\mathbf{p}^t$  按式 (6) 改进:

$$\bar{\mathbf{p}}^t = \sum_{j=1}^m \mathbf{p}^t_j / m \quad (6)$$

其中,  $m = N/4$  (取整),  $\mathbf{p}^t_1, \mathbf{p}^t_2, \dots, \mathbf{p}^t_m$  为适应度值从小到大排序的前 1/4 粒子对应的个体极值,改进后的速度更新公式为:

$$\mathbf{v}^{t+1} = \omega \mathbf{v}^t + c_1 r_1 (\bar{\mathbf{p}}^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{p}^t_g - \mathbf{x}_i^t) \quad (7)$$

### 2.4 概率控制

引入一个随机决策数  $P$ ,  $P$  是以迭代次数  $t$  为变量构建的函数,在迭代初期取大值,在迭代后期取小值,如式 (8) 所示:

$$P = 0.1 + 0.9e^{-10t/G_{\max}} \quad (8)$$

若  $rand(0, 1) \geq P$ , 则进行交叉变异等操作,否则不进行.

### 2.5 算法流程

对 IPSO 算法流程描述如下,图 1 给出了 IPSO 算法流程图.

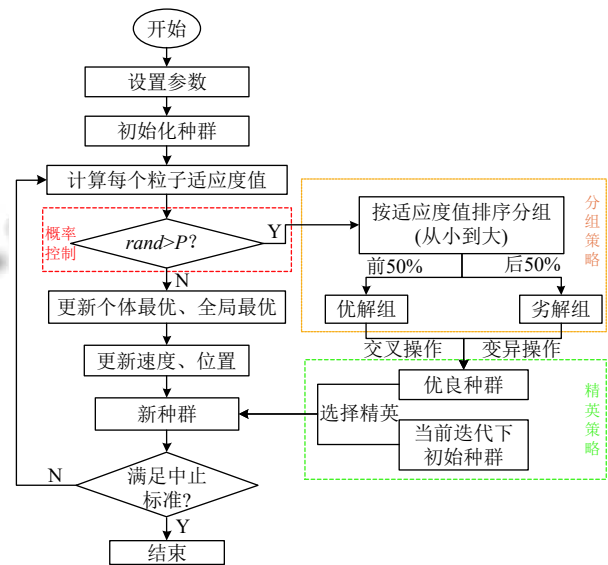


图 1 IPSO 算法流程图

- Step 1. 设置算法参数;
- Step 2. 随机初始化种群;
- Step 3. 计算每个粒子适应度值;
- Step 4. 判断是否进入交叉变异操作,若  $rand(0, 1) \geq P$ , 进入 Step 5, 否则跳至 Step 7;

Step 5. 根据适应度值按升序排列粒子, 前 50% 个粒子被定义为优解组, 其余为劣解组, 优解组进行交叉操作, 劣解组进行变异操作, 产生新的种群;

Step 6. 将经过交叉变异后新产生的种群与当前迭代次数下的初始种群合并形成扩大种群, 按适应度值升序排列, 选出前一半粒子作为精英种群用于更新迭代;

Step 7. 按式 (6) 更新个体最优, 更新全局最优;

Step 8. 更新粒子位置和速度;

Step 9. 判断是否满足迭代中止标准, 若迭代次数达到最大迭代次数  $G_{max}$ , 算法停止; 否则跳至 Step 3.

### 3 算法仿真测试

#### 3.1 实验环境及参数设置

实验环境为: Matlab R2014b, CPU 为 i7-6500U@2.5 GHz, 选取基本 PSO<sup>[1]</sup>, BreedPSO<sup>[12]</sup>, SimuAPSO<sup>[14]</sup> 三种算法与本文提出的 IPSO 算法进行比较. 算法参数的设置同原文献相同, 对于 BreedPSO, 杂交概率取 0.9, 杂交池的大小取 0.2; 对于 SimuAPSO, 退火常数取 0.5. 各优化方法共有参数均保持一致, 种群规模取 400, 最大迭代次数为 200, 学习因子均取 2.

#### 3.2 测试函数

从 GEATbx 中选取 4 个标准测试函数, 测试函数维数取 10 维, 分别对 IPSO 算法和其他改进 PSO 算法性能进行了比较. 其中  $f_1$ 、 $f_2$  为单峰函数,  $f_1$  用于检验算法的收敛速度和精度;  $f_2$  最优位置是在一个狭长的, 抛物线形的扁平山谷内, 可认为拥有无数个局部极值,  $f_3$ 、 $f_4$  为多峰函数, 其二维函数图像如图 2、图 3 所示, 其局部极值的个数随维数的增加成几何级增加,  $f_2$ 、 $f_3$ 、 $f_4$  主要用于检验算法跳出局部极值, 全局搜索的能力, 同时也能检验其收敛速度和精度.  $f_1$ 、 $f_3$ 、 $f_4$  函数的全局最优均为: 当  $x_i = 0$ , ( $i = 1 : n$ ),  $f(x) = 0$ ;  $f_2$  的全局最优为: 当  $x_i = 1$ , ( $i = 1 : n$ ),  $f(x) = 0$ . 为消除随机性对算法性能带来的影响, 对每个测试函数采用不同 PSO 算法进行 100 次独立运算, 得到各测试函数迭代完成后的适应度值, 以 100 次适应度值的均值和方差作为评价指标, 均值能最直观反映优化结果的优劣, 检验算法的收敛速度和精度, 而方差从不确定性量化的角度反映优化结果的稳定性.  $f_1$  为单峰球函数,  $x_i$  越接近于 0, 适应度值越小, 优化速度及精度越高, 设置成功率来评判优化性能, 表 1 显示了不同优化方法对  $f_1$  优化的结果. 函数  $f_2$ 、 $f_3$ 、 $f_4$  全局最优附近含有较多局部峰值,

若最终的适应度值小于  $10^{-5}$ , 则认为成功搜索到全局最优解. 各种不同优化算法对测试函数  $f_2$ 、 $f_3$ 、 $f_4$  优化结果如表 2 所示. 图 (4)~图 (7) 分别为几种不同 PSO 算法在对函数  $f_1$ ~ $f_4$  寻优时的平均适应度值收敛曲线, 能更全面直观展示算法的迭代优化过程.

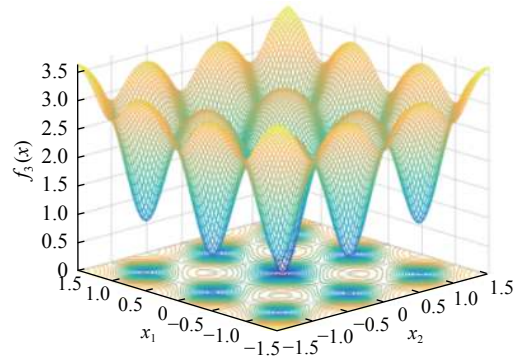


图 2 Ackley's Path 二维函数图像

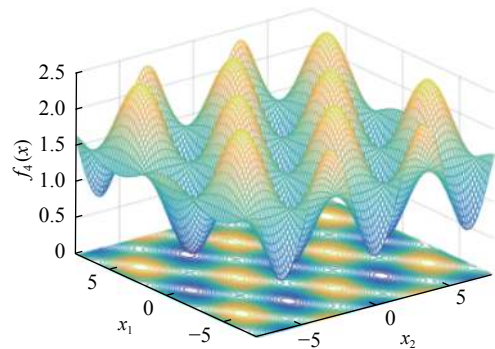


图 3 Griewank 二维函数图像

(1) Sphere 球函数:  $x \in [-5.12, 5.12]$

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad (9)$$

(2) Rosenbrock 函数:  $x \in [-2.048, 2.048]$

$$f_2(x) = \sum_{i=1}^{n-1} \left( 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right) \quad (10)$$

(3) Ackley's Path 函数:  $x \in [-1.5, 1.5]$

$$f_3(x) = -5 \cdot e^{-0.2 \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}} - e^{\frac{\sum_{i=1}^n \cos(2\pi x_i)}{n}} + 5 + e^1 \quad (11)$$

(4) Griewank 函数:  $x \in [-8, 8]$

$$f_4(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (12)$$

表1 测试函数  $f_1$  优化的结果

优化算法	种群规模	函数维度	计算100次		成功率(%)			运行时间(s)
			均值	方差	Obj<10 <sup>-20</sup>	Obj<10 <sup>-30</sup>	Obj<10 <sup>-40</sup>	
PSO	400	10	5.69E-12	3.35E-23	0	0	0	0.799
BreedPSO	400	10	3.37E-19	1.61E-30	7	0	0	5.188
SimuAPSO	400	10	4.02E-19	1.64E-37	4	0	0	2.565
BEPSO	400	10	3.03E-25	4.25E-35	100	77	45	3.106

表2 测试函数  $f_2$ 、 $f_3$ 、 $f_4$  优化的结果

优化算法	种群规模	测试函数	函数维度	最优解位置	计算100次		成功率(%)	运行时间(s)
					均值	方差	Obj<10 <sup>-5</sup>	
PSO	400	$f_2$	10	(1, ..., 1)	7.307	429.521	23	0.814
BreedPSO					4.484	244.581	27	6.416
SimuAPSO					5.829	233.654	25	2.613
BEPSO					0.823	5.681	56	3.346
PSO	400	$f_3$	10	(0, ..., 0)	0.556	0.034	5	0.431
BreedPSO					0.347	0.052	20	10.34
SimuAPSO					0.448	0.063	24	3.387
BEPSO					1.33E-02	4.40E-03	96	3.714
PSO	400	$f_4$	10	(0, ..., 0)	0.195	0.017	0	2.266
BreedPSO					0.178	0.015	1	29.44
SimuAPSO					0.186	0.018	0	3.658
BEPSO					8.02E-04	9.75E-06	90	4.454

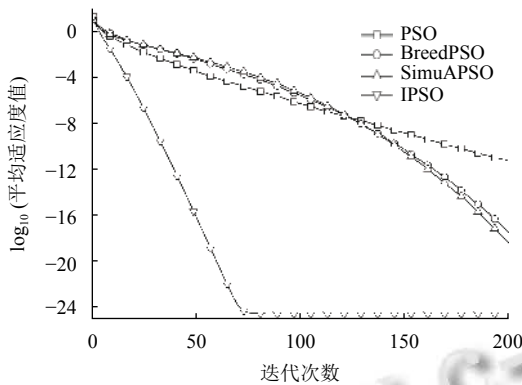


图4  $f_1$  平均适应度值的对数迭代曲线

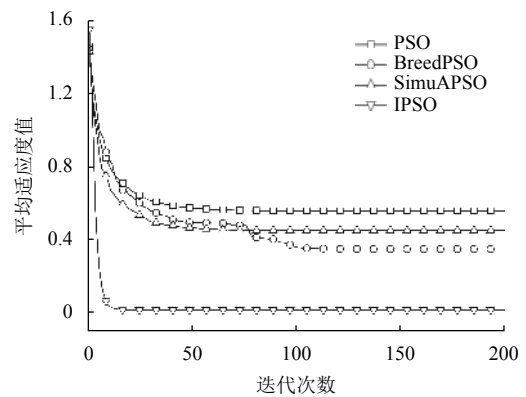


图6  $f_3$  平均适应度值的迭代曲线

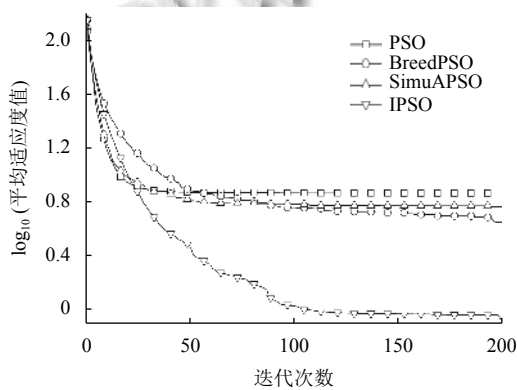


图5  $f_2$  平均适应度值的对数迭代曲线

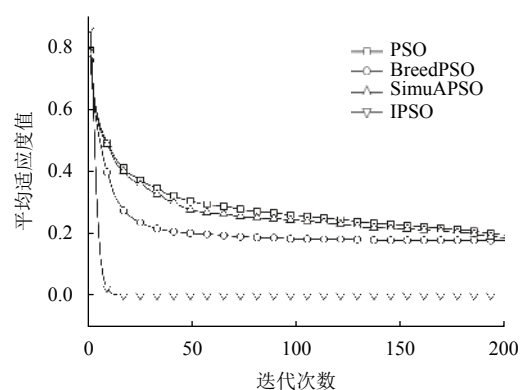


图7  $f_4$  平均适应度值的迭代曲线

### 3.3 仿真结果分析

从表1和表2可知, IPSO算法分别对4个测试函数进行100次计算得到的适应度值均值和方差均小于其他PSO算法, 表明IPSO算法收敛速度快, 搜索精度高, 稳定性好. 对于测试函数 $f_1$ , IPSO算法计算得到适应度值均值为 $3.03E-25$ , 相比其他PSO算法, 搜索精度显著提高, 在最佳目标成功率上, 100%达到 $10^{-20}$ 数量级, 优化到 $10^{-40}$ 以下成功率达到45%, 远远优于其他PSO算法. 另外, 在相同条件下运行时间, IPSO算法要比基本PSO耗时长, 与SimuAPSO相当, 比BreedPSO耗时短, 从取得的优化效果来看, 增加的时间完全可以接受.

图4—图7中平均适应度值收敛曲线显示了IPSO算法下降速度快, 达到稳定时迭代次数少, 稳定时平均适应度值远低于其他PSO算法, 进一步证实IPSO算法具有收敛速度快、精度高的特点. 从表2可知, IPSO算法在提高收敛速度和精度的同时, 跳出局部极值成功搜索到全局最优解的能力显著提升, 对于测试函数 $f_2$ , 搜索到全局最优 $(1, \dots, 1)$ 附近位置, IPSO算法成功率达到56%, 而其他PSO算法均小于30%, 对于测试函数 $f_3$ , 搜索到全局最优 $(0, \dots, 0)$ 附近位置, IPSO算法成功率高达96%, 而其他PSO算法均小于30%. 测试函数 $f_4$ 对比于 $f_3$ , 其局部极值与全局最优位置对应的适应度值相差很小且局部极值数量庞大, 因此很难找到全局最优位置, IPSO算法找到最优解 $(0, \dots, 0)$ 位置的成功率为90%, 而其他PSO算法均很难找到最优解位置.

## 4 结论

本文针对PSO算法容易陷入局部极值及进化后期收敛速度慢、精度低等缺点, 提出了一种融合多种策略的改进粒子群算法(IPSO). 该算法采取了分组控制、精英选择、改变学习模式、引入概率等策略, 通过实验仿真结果证明了IPSO算法同其他PSO算法相比, 收敛速度快、求解精度高且容易跳出局部最优解. IPSO算法的全局探索和局部挖掘能力均得到明显提升, 较为有效地兼顾了全局与局部寻优.

### 参考文献

- Kennedy J, Eberhart RC. Particle swarm optimization. Proceedings of International Conference on Neural Networks (ICNN'95). Perth, WA, Australia. 1995. 1942–1948.
- 夏学文, 刘经南, 高柯夫, 等. 具备反向学习和局部学习能力的粒子群算法. 计算机学报, 2015, 38(7): 1397–1407. [doi: 10.11897/SP.J.1016.2015.01397]
- Verma R, Mehra R. PSO algorithm based adaptive median filter for noise removal in image processing application. International Journal of Advanced Computer Science and Applications, 2016, 7(7): 92–98.
- Bian Q, Zhao KR, Wang XM, et al. System identification method for small unmanned helicopter based on improved particle swarm optimization. Journal of Bionic Engineering, 2016, 13(3): 504–514. [doi: 10.1016/S1672-6529(16)60323-2]
- 王荣, 白尚旺, 党伟超, 等. 粒子群退火算法优化的BP神经网络及其应用. 计算机系统应用, 2020, 29(1): 244–249. [doi: 10.15888/j.cnki.csa.007220]
- 秦世强, 胡佳, 曹鸿猷, 等. 基于试验数据的大跨度拱桥有限元模型修正. 中国公路学报, 2019, 32(7): 66–76.
- 吴静, 罗杨. 动态调整惯性权重的粒子群算法优化. 计算机系统应用, 2019, 28(12): 184–188. [doi: 10.15888/j.cnki.csa.007162]
- 张继荣, 张天. 基于改进粒子群算法的PID控制参数优化. 计算机工程与设计, 2020, 41(4): 1035–1040.
- 李龙涛, 张效见. 一种新的自适应惯性权重混沌PSO算法. 计算机工程与应用, 2018, 54(9): 139–144. [doi: 10.3778/j.issn.1002-8331.1612-0093]
- 谭阳, 唐德权, 全惠云. 一种排异竞争的粒子群优化算法. 系统仿真学报, 2011, 23(12): 2635–2640, 2646.
- Liang JJ, Qin AK, Suganthan PN, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281–295. [doi: 10.1109/TEVC.2005.857610]
- Jiang JH, Li X, Deng ZH, et al. Control-oriented dynamic model optimization of steam reformer with an improved optimization algorithm. International Journal of Hydrogen Energy, 2013, 38(26): 11288–11302. [doi: 10.1016/j.ijhydene.2013.06.103]
- Xia ZY, Li AQ, Li JH, et al. FE model updating on an in-service self-anchored suspension bridge with extra-width using hybrid method. Applied Sciences, 2017, 7(2): 191. [doi: 10.3390/app7020191]
- 刘波, 张焰, 杨娜. 改进的粒子群优化算法在分布式电源选址和定容中的应用. 电工技术学报, 2008, 23(2): 103–108. [doi: 10.3321/j.issn:1000-6753.2008.02.017]
- 张庭场, 耿光飞. 基于改进粒子群算法的中压配电网无功优化. 电网技术, 2012, 36(2): 158–162.
- Darwin C. On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. London: John Murray, Albemarle Street, 1859.