DOI：10. 13471/j. cnki. acta. snus. 2020. 09. 07. 2020A047

混合分层自主学习量子粒子群优化算法*

王文丰[1,2]，徐灯[2]，傅晶[2]，韩龙哲[1,2]，方宗华[2]，董健华[2]，章香[2]

1. 南昌工程学院江西省水信息协同感知与智能处理重点实验室，江西 南昌 330099
2. 南昌工程学院信息工程学院，江西 南昌 330099

摘　要：针对粒子群优化算法容易陷入局部最优、收敛精度不高以及收敛速度较慢的问题，本文提出一种混合分层自主学习量子粒子群优化算法HHQPSO。首先，根据粒子适应度值和迭代次数将种群动态划分为三个阶层：上、下两层粒子分布较少，分别采用局部学习模型和全局学习模型，以增强粒子多样性；中层粒子分布较多，采用混合自适应量子学习模型。其次，在混合量子模型中提出改进差分策略以更新粒子的随机位置，并引入Levy飞行策略以提高算法的收敛精度和收敛速度。最后，分别在9个典型测试函数上对6种改进粒子群算法进行仿真对比实验。实验结果表明，HHQPSO算法在收敛精度、速度和稳定性上均有着较为明显的优势，特别适用于多峰函数寻优。

关键词：量子粒子群；差分策略；Levy飞行策略；自主学习

中图分类号：TP393　　**文献标志码：**A　　**文章编号：**0529－6579（2021）06－0161－08

Hybrid hierarchical autonomous learning quantum particle swarm optimization

WANG Wenfeng[1,2], *XU Deng*[2], *FU Jing*[2], *HAN Longzhe*[1,2], *FANG Zonghua*[2], *DONG Jianhua*[2], *ZHANG Xiang*[2]

1. *Jiangxi Province Key Lab of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang 330099, China*
2. *School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China*

Abstract: In order to solve the problem that particle swarm optimization is easy to be trapped in local optimum, low convergence precision and slow convergence speed, a hybrid hierarchical autonomous learning quantum particle swarm optimization (HHQPSO) is proposed. Firstly, the population is dynamically divided into three layers according to the fitness value of particles and the number of iterations. Since the particles of upper and lower layers are less distributed, local learning model and global learning model are respectively used to increase the diversity of particles. While the particles of middle layer are more distributed, the hybrid adaptive quantum learning model is adopted. Secondly, an improved differential strategy is proposed to update the random positions of particles in the hybrid quantum model, and the Levy flight strategy is introduced to improve the convergence accuracy and speed of the algorithm. Finally, six improved particle swarm optimization algorithms are compared on 9 typical test functions. Experimental results show that the HHQPSO algorithm has obvious advantages in convergence accuracy, speed and stability, especially suitable for multi-peak search function optimization.

Key words: quantum particle swarm; differential strategy; Levy flight strategy; autonomous learning

* **收稿日期：**2020－09－07　　**录用日期：**2020－12－08　　**网络首发日期：**2021－01－06
基金项目：国家自然科学基金(61962036,61561035)；2020年国家大学生创新创业训练计划(202011319014)
作者简介：王文丰(1983年生)，男；**研究方向：**分布式计算、云存储及智能计算；E-mail：Wangwf@nit. edu. cn

粒子群算法因借鉴鸟群集体捕食的特性，便于理解、参数少且易实现，在科学研究与工程实践中备受关注，譬如在路径规划、图像处理[1]和洪水预报[2]等众多领域中都得到了广泛应用。然而，PSO算法却也存在着一些不足之处，如：前期因粒子的搜索范围固定易陷入局部最优使其早熟、后期因粒子的搜索方式单一而导致粒子收敛速度较慢从而无法保证搜索精度等。

针对上述问题，国内外专家学者研究并提出了多种改进方法和策略。孙辉等[3]提出从维度值角度出发来提高种群的多样性，为得到变异的新维度值，首先选取优质粒子，然后将粒子随机的散落到不同的维度空间中，用列维飞行的方式从一个维度移动到另一个维度中，最终获得了变异新维度。张鹏威等[4]为了加强离散型二进制板块中粒子的全局搜索能力，采用了正弦映射和扩张算子的方法。Chen等[5]采用基于位置的学习方法对种群进行初始化，并引入正余弦加速度系数来控制局部搜索范围和全局最优解收敛速度。Hakl等[6]利用Levy飞行策略改变固定步长的特性与PSO算法结合来提高粒子的收敛精度。Liang等[7]为加强粒子间信息交流能力使其多个小种群频繁重组交流并使用拟牛顿法来增强局部搜索能力。上述研究对PSO算法在种群跳出局部最优、粒子多样性和参数设置等方面进行了改进，在一定程度上提高了算法的整体性能，但是依然有大部分的算法都有收敛速度慢，精确度低，过早成熟的各种问题，这些问题最常发生在高维函数优化问题上。

文献[8]提出基于分层自主学习的粒子群优化算法HCPSO。该算法在一定数量种群粒子迭代过程中对种群进行阶层划分，并对划分出的各粒子阶层匹配合适的学习策略，从而提高粒子的学习效率，但同时由于中层粒子数较多，因此限制了粒子搜索能力。本文认为可以使用混合分层自主学习量子粒子群优化算法（HHQPSO）来解决问题，在保证粒子学习自主性的同时提升了算法全局搜索能力，提高了算法的稳定性以及收敛速度和精度，且在高维函数上有较好的适应性。

1 量子粒子群算法

量子粒子群算法[9]是信息学科交叉融合产生的一种新型粒子群算法，其引入量子力学的特性增强粒子间的信息交流，将粒子的搜索范围明确化，从而使量子的搜索能力提升。量子粒子群算法相对于传统粒子群算法来说多了一个量子空间概念，增加了一个引力场δ势阱，在引力场的作用之下，所有的粒子向某一中心点靠近。在此过程中粒子的速度和位置出现的概率是确定的，而运动轨迹则可以使用波函数ψ来表示。因为有δ势阱引力场的作用，所以粒子的运动方向就是朝着吸引点不断地前进，粒子可以通过薛定谔方程来确定概率密度，而粒子的运动轨迹可以通过蒙特卡洛随机模拟方式来描述，表示如下：

$$x_{id}^{t+1} = p_i^t \pm \frac{L_{id}(t)}{2} \ln\left(1/u_{id}^t\right), \tag{1}$$

其中x_{id}^{t+1}表示的是粒子迭代次数在第t+1次时i在第d维的位置；p_i^t表示的是粒子迭代次数在第t次时粒子i的随机位置，将相应的数值代入到以下的公式中

$$p_i^t = \phi_d(t)\, p_{id}^t + [1 - \phi_d(t)]\, p_{gd}^t , \tag{2}$$

其中在(0,1)之间随机数用ϕ_d表示，p_{id}^t为粒子i在d维的最优位置，p_{gd}^t为种群的最优位置。

式(1)中，±的选择取决于u_{id}的大小，取值范围是0~1之间，正负选择以0.5为界限，小于0.5的为负，大于0.5的为正；$L_{id}(t)$为势阱长度，计算公式如下

$$L_{id}(t) = 2\alpha(t) \left| C_d^t - x_{id}^t \right|, \tag{3}$$

其中$\alpha(t)$为压缩因子[10]；C_d^t为平均最优位置，即

$$C_d^t = \frac{1}{N} \sum_{i=1}^{N} P_i^t . \tag{4}$$

将式(3)代入式(1)中即可得到新的粒子进化方程

$$x_{id}^{t+1} = p_i^t \pm \alpha(t) \left| C_d^t - x_{id}^t \right| \times \ln(1/u_{id}^t). \tag{5}$$

2　混合分层自主学习量子粒子群优化算法

2.1　混合量子粒子模型

2.1.1　粒子随机位置更新策略的改进　差分进化是根据生物进化理论，经过杂交、交叉和变异等行为之后，进化成能够适应当前环境的优质粒子，进化之后的粒子具有更好的鲁棒性。粒子的随机位置会随时改变，自然就导致了局部搜索能力较低的问题，为了解决这一问题，本文在差分策略的基础上加入了缩放因子的方法，其计算公式如下

$$p_i^{t+1} = p_{(r0)gd}^t + F*[\,p_{(r1)gd}^t - p_{(r2)gd}^t\,], \tag{6}$$

其中F表示的是缩放因子，会受到当前粒子适应度值的影响，具体的计算方法如下：

$$F = \begin{cases} F_{\min} - \dfrac{(F_{\max} - F_{\min}) \times (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leqslant f_{\text{avg}}; \\ F_{\max}, & f > f_{\text{avg}}, \end{cases} \tag{7}$$

其中f表示的是当前粒子最优适应度值。

把小f和适应度均值对比，若f小于均值，就需要减少缩放因子，这样在均值的附近就会分布更多的粒子，从而提高搜索的能力；相反，如果f大于均值，需要增大缩放因子，扩张了缩放因子，提高最优质的获取几率。查阅相关文献之后发现文献[11]，本文选取$F_{\max}$和$F_{\min}$的值分别为0.9和0.4。

2.1.2　粒子位置更新策略的改进　Levy飞行策略[12,13]是效仿生物寻找新食物的一种行为，其随机的出现在某一地点且到下一地点的移动距离不确定，因此这一特征可以防止粒子容易陷入局部最优。粒子群中粒子的位置在更新的过程中经常会有固定步长变化的问题存在，所以本文用Levy飞行策略来进一步的扩大搜索范围，提高局部搜索的能力。

Levy飞行策略的分布方式是根据Levy规律来完成的，通过化简与Fourier变换后，概率密度函数如下所示

$$\text{Levy}\sim\mu = t^{-\lambda}; \quad 1 < \lambda < 3. \tag{8}$$

式(8)中的概率分布是有宽尾的，尾翼的宽度比高斯分布和柯西分布要更大一些，所以从扰动能力上来看，这种概率分布要更强一点。通常使用Mantegna提出的Levy飞行路径表达式：

$$\text{Levy}(\lambda) = \mu/|\upsilon|^{1/\beta} \tag{9}$$

式中参数β的取值范围是0~2，一般情况下是1.5；参数μ和ν是服从正态分布的随机数，即$\mu \sim N(0, \sigma_\mu^2)$、$\nu \sim N(0,\sigma_\upsilon^2)$，其中标准差$\sigma_\mu$和$\sigma_\upsilon$取值满足下式

$$\begin{cases} \sigma_\mu = \left\{ \dfrac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \times \beta \times 2^{(\beta-1)/2}} \right\}^{1/\beta}, \\ \sigma_\upsilon = 1. \end{cases} \tag{10}$$

可以计算出飞行路径Levy(λ)，而粒子位置更新公式如下所示

$$\begin{cases} x_{id}^{t+1} = p_i^t + A + B, & u_{id}^t > 0.5; \\ x_{id}^{t+1} = p_i^t + A - B, & u_{id}^t \leqslant 0.5, \end{cases} \tag{11}$$

其中

$$\begin{cases} A = \text{Levy}(\lambda)(x_{id}^t - p_{gd}^t), \\ B = \alpha(t)\left| C_d^t - x_{id}^t \right| \times \ln(1/u_{id}^t). \end{cases}$$

2.2　分层自主学习

2.2.1　分层核心思想　算法在一定数量种群粒子迭代过程中对种群进行阶层划分，并对划分出的各粒子阶层选择合适的学习方法。在算法初始阶段中，种群粒子是按照正态分布来确定搜索范围的，最终集合的位置是唯一的，所以在最优的位置，粒子一般都比较稀少，同时也很难找到，只有很少部分的最优位置的粒子真正地能够达到最优的位置上，而且也有一些粒子因为找不到最优的位置，所以在其他的地方

徘徊。所以把处于最优位置附近的少数例子看成是上层粒子，为了提高局部范围的搜索精度，使用局部学习模型；距离最优位置较远的粒子叫做中层粒子，它通过混合量子粒子模型连接了群体信息中的粒子，这不仅仅是提高了局部的搜索精度，而且对全局的搜索精度也能起到优化的作用；而下层粒子指的是距离最优位置最远的部分，而粒子的分布规律和位置的选择是根据正态分布来完成的，可以有效地提高粒子搜索最优位置的速度。这种分层方式不仅能够使种群类型更加丰富，同时粒子的收敛速度也得到了提升。

2.2.2 *粒子分层策略及学习模型* 记f_i表示第i个粒子适应度值，按照粒子适应度值大小，对粒子按照从小到大的方式排序，适应度值越接近最优位置，那么就越小，将更新之后的粒子序列记为$X=\{x_1,x_2,\cdots,x_N\}$，N表示的是粒子总数。下界和上界分别用low和up来表示，此时种群动态的三个阶层就由图1所示。

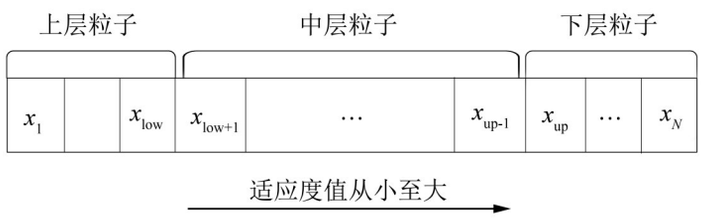

图1 粒子分层示意图

Fig. 1 Particle classification diagram

根据参考文献[8]的内容，选取合适的low和up值，取值公式动态更新如下

$$\begin{cases}\text{low} = \text{count}_{\text{start}} + N*(0.5 - t/T)^3,\\ \text{up} = \text{count}_{\text{end}} + N*(0.5 - t/T)^3,\end{cases} \tag{12}$$

其中$\text{count}_{\text{start}}$和$\text{count}_{\text{end}}$分别为$N/8$和$7N/8$，$t$和$T$分别表示当前迭代次数和最大迭代次数。

据此，粒子分层策略和学习模型可以通过以下一些方式来表达。

1） 如果$x_i \in \{x_1, x_2, \cdots, x_{\text{low}}\}$，表示的是上层粒子，速度更新学习模型为

$$\begin{cases}v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1 (p_{id} - x_{id}^{t}) + c_2 r_2 (\overline{\text{up}} - x_{id}^{t}),\\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1},\end{cases} \tag{13}$$

其中$\overline{\text{up}}$表示上层粒子位置期望值。

2） 如果$x_i \in \{x_{\text{low}+1}, x_{\text{low}+2}, \cdots, x_{\text{up}-1}\}$，表示的是中层粒子，使用混合量子粒子位置更新学习模型。

3） 如果$x_i \in \{x_{\text{up}}, x_{\text{up}+1}, \cdots, x_N\}$，表示的是下层粒子，速度更新学习模型为

$$x_{id}^{t+1} = \text{rand}*\left(x_{id}^{u} - x_{id}^{l}\right) + x_{id}^{l}, \tag{14}$$

其中x_{id}^{u}，x_{id}^{l}分别表示粒子i上界和下界的位置。

2.3 算法关键步骤

Step 1 粒子的参数设定，设置好随机初始化种群中N个粒子的数值；

Step 2 粒子适应度值计算，确定并保存好局部和全局最优粒子的位置；

Step 3 将适应度值按照升值的方式排序，然后利用式(12)计算出粒子分层的下界和上界low、up值，将种群分成了上中下三层；

Step 4 粒子学习模型的选择。参考粒子所在阶层自适应的情况来选择相应的学习模式：分布在上层阶段的粒子，通过式(13)来计算粒子的速度和位置，找到最优粒子的位置和适应度值，然后按照要求保存好；分布在中层阶段的粒子，使用式(6)和式(11)确定好随机点的位置和粒子更新之后的位置，将最优的适应度值和位置确定下来，并且保存；处于下层阶段的粒子，利用式(14)忧患粒子的位置，确定最优粒子的适应度值和位置，并且保存；

Step 5 对比这三个阶层的最优适度值，选择最小的，如果这一个粒子的全局最优值要比之前的要好，那么就保留更新之后的适应度值和位置，如果不如迭代前的适应度值，那么保留的是迭代前的粒子是适

应度值以及位置；

Step 6　继续根据 Step 3~Step 5 的步骤来操作，直到迭代次数达到最大值，或者达到了其他的算法终止需求，就可以输出，此时是最优解。

3　仿真实验及分析

3.1　测试函数和参数设置

本文选取了单峰和多峰函数共 9 个，都是 CEC 2005 基准函数中的测试函数，如表 1 所示。算法的参数设置：种群规模为 40，c=1.5~3，c_1=0.5~2.5，c_2=0.5~2.5 .

表 1　基准测试函数

Table 1　Benchmark function

Name	Function	Range	Optimum
Sphere	$f_1(x)=\sum_{i=1}^{D}x_i^2$	[−100，100]	0
Schwefel 2.22	$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[−10，10]	0
Schwefel 1.2	$f_3(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2$	[−100，100]	0
Schwefel 2.21	$f_4(x)=\max\{\lvert x_i\rvert,1\leqslant i\leqslant D\}$	[−100，100]	0
Rosebrock	$f_5(x)=\sum_{i=1}^{D}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i^2\right)^2\right]$	[−5，10]	0
Step	$f_6(x)=\sum_{i=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$	[−100，100]	0
Noise	$f_7(x)=\sum_{i=1}^{D}ix_i^4+\mathrm{random}[0,1)$	[−1.28，1.28]	0
Rastrigin	$f_8(x)=\sum_{i=1}^{D}\left[x_i^2-10\cos 2\pi x_i+10\right]$	[−5.12，5.12]	0
Griewank	$f_9(x)=\frac{1}{4\,000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[−600，600]	0

3.2　仿真实验及结果分析

PSO 算法选择了最具代表性的 5 个，分别是 HPSO-TVC[14]、CLPSO[15]、LFPSO[6]、DMS-PSO[7]和 RPSOLF[16]，参考文献[17]是对比这些算法的结果之后得到的结果。本文选取的评价指标为最优适应值的平均值 Mean 和标准差 SD，实验结果见表 2。

从表中可以看出，如果评估次数固定，那么最优位置一般是在f_1、f_2、f_3、f_4、f_5、f_6、f_8、f_9中，求得的最优值精度也是最高的，这也就意味着粒子经过 Levy 飞行策略的优化之后，粒子能够抵抗干扰的能力得到了大幅度的增加。从标准差的稳定线上来看，LFPSO 要更好，但是f_5(单峰)f_6、f_8、f_9(多峰)的适应度值为 0，为最佳的理论状态。LFPSO 在多峰函数f_9上求得的最优值精确度比 HPSO-TVAC 和 DMSPSO 算法求解要高出 14 个数量级，但是能够达到最优值的测试函数，也只有f_6，而 RPSOLF 算法求解得到的理论最优质值在多峰函数f_6、f_8、f_9上都可以得出，这也就是说 LFPSO 算法只有 Levy 飞行策略在高危复杂度问题计算的过程中是不可行的。RPSOLF 通过三个测试函数求出理论最优质，但是从精确度上来看，HPSO-TVAC 和 DMSPSO 相对于函数f_1、f_2来说要更高一些。本文的最优值求解是通过函数f_5、f_6、f_8、f_9求出的，而最优的精确度最好的是函数f_1、f_3，这也就说明了本文所使用的算法分层策略更能够求出最优粒子。

通过寻优曲线图可以直接对比各算法在不同函数的收敛趋势，本文把上述 9 个测试函数在 6 种算法上做收敛情况对比，如图 2 所示，图中 FEs 代表的是评估的次数，而每次评估之后得到的适应值用 fitness 表示。

从迭代曲线图中可以了解到，在收敛速度和精度的问题上，f_1、f_2、f_3、f_4(单峰)和f_7(多峰)的效果是最

表2 算法实验结果对比

Table 2 Comparison of experimental results

Function		HPSO-TVAC	DMSPSO	CLPSO	LFPSO	RPSOLF	HHQPSO
f_1	Mean	2. 83E-33	2. 65E-31	1. 58E-12	4. 69E-31	1. 35E-22	2. 60E-143
	SD	3. 19E-33	6. 25E-31	7. 70E-13	2. 50E-30	6. 58E-21	9. 07E-72
f_2	Mean	9. 03E-20	1. 57E-18	2. 51E-08	2. 64E-17	5. 55E-19	3. 84E-71
	SD	9. 58E-20	3. 79E-18	5. 84E-09	6. 92E-17	3. 58E-18	1. 10E-35
f_3	Mean	5. 68E-02	2. 89E-02	2. 56E-01	2. 65E-01	8. 05E-14	1. 98E-136
	SD	5. 63E-01	6. 32E-01	6. 38E-01	9. 32E-01	1. 57E-12	2. 50E-68
f_4	Mean	3. 25E-02	6. 85E-03	5. 65E-02	3. 55E-03	6. 67E-11	9. 79E-73
	SD	3. 56E-08	6. 58E-06	2. 35E-08	3. 84E-02	1. 20E-09	1. 71E-36
f_5	Mean	2. 39E+01	4. 16E+01	3. 82E-04	2. 38E+01	2. 52E+01	0. 00E+00
	SD	2. 65E+00	3. 03E+01	1. 28E-07	3. 17E-01	2. 68E+00	0. 00E+00
f_6	Mean	0. 00E+00	0. 00E+00	0. 00E+00	0. 00E+00	0. 00E+00	0. 00E+00
	SD	0. 00E+00	0. 00E+00	0. 00E+00	0. 00E+00	0. 00E+00	0. 00E+00
f_7	Mean	9. 82E-02	1. 45E-02	5. 85E-03	2. 41E-03	2. 58E-02	1. 83E-04
	SD	3. 26E-02	5. 05E-03	1. 11E-03	8. 07E-04	3. 21E-01	1. 07E-02
f_8	Mean	9. 43E+00	2. 72E+01	9. 09E-05	4. 54E+00	0. 00E+00	0. 00E+00
	SD	3. 48E+00	6. 02E+00	1. 25E-04	1. 03E+01	0. 00E+00	0. 00E+00
f_9	Mean	9. 75E-03	6. 21E-03	9. 02E-09	8. 14E-17	0. 00E+00	0. 00E+00
	SD	8. 33E-03	8. 14E-03	8. 57E-09	4. 46E-16	0. 00E+00	0. 00E+00

好的，这些函数得到的结果会更优（图2的（a）~（d）、（g））。虽然CLPSO的收敛速度在f_5（单峰）上表现的非常好，但是也无法比得上本次的算法（图2的（e）），所以下层粒子的最优位置，寻找速度最快的可以使用全局学习模型，并且是使用多峰函数f_6、f_8、f_9求解，本次上最优理论值的过程中，当评估次数在5 000次之前就已经无限接近了（图2的（f）、（h）、（i）），说明本分层策略的解决效率是最好的，并且中层粒子按照混合量子学习模型和改进位置之后，粒子之间的信息交流得到了进一步的加强和改善，使得粒子在前期不易陷入早熟，提高全局搜索能力从而找到更好的解。

如前所述，HHQPSO结合了HCPSO和混合量子粒子策略，通过实践表明，这种模型能够使粒子的搜索特性得到加强，本文同时测试了单峰函数Sphere和多峰函数Griewank，并且在实验过程中出现的结果记录下来，进行对比。实验中的参数设置按照上述的要求来完成，得到了图3的寻优曲线图。

由图可知，单峰函数f_1中，两个算法在中前期的收敛情况基本一致，而在评估2. 6万次后，本文算法在收敛精度与速度上明显优于HCPSO算法，说明文中提出的混合量子粒子模型有效解决了粒子陷入局部最优位置的问题，因此该模型提高了粒子勘探能力；在多峰函数f_9中，HHQPSO算法在评估前期4 000次左右时就求解得到理论最优值且收敛速度极快，反观HCPSO算法的收敛情况在前期也保持稳定的收敛速度，但在评估5 000次左右时收敛速度变慢最后在评估2. 5万次时陷入了局部最优。通过实验表明，从全局搜索能力、收敛速度、精确度上来看，本文所使用的算法要更好一些，而且高维函数的开发能力要更好一点。

4 结 语

为了找到PSO粒子分布过程中早熟，过于单一，迭代后期的收敛速度变缓的各种问题的解决方法，本文通过以下两点方式来进行改进：第一，采用混合量子粒子自适应学习模型，中层粒子的数量和种类都得到了优化和充实，粒子的抗干扰能力也得到了进一步的加强，而且相对于普通的粒子来说，可以更快地获得最优解；第二，使用的是粒子自主分层学习策略，根据距离最优位置的适应度值将种群分为上

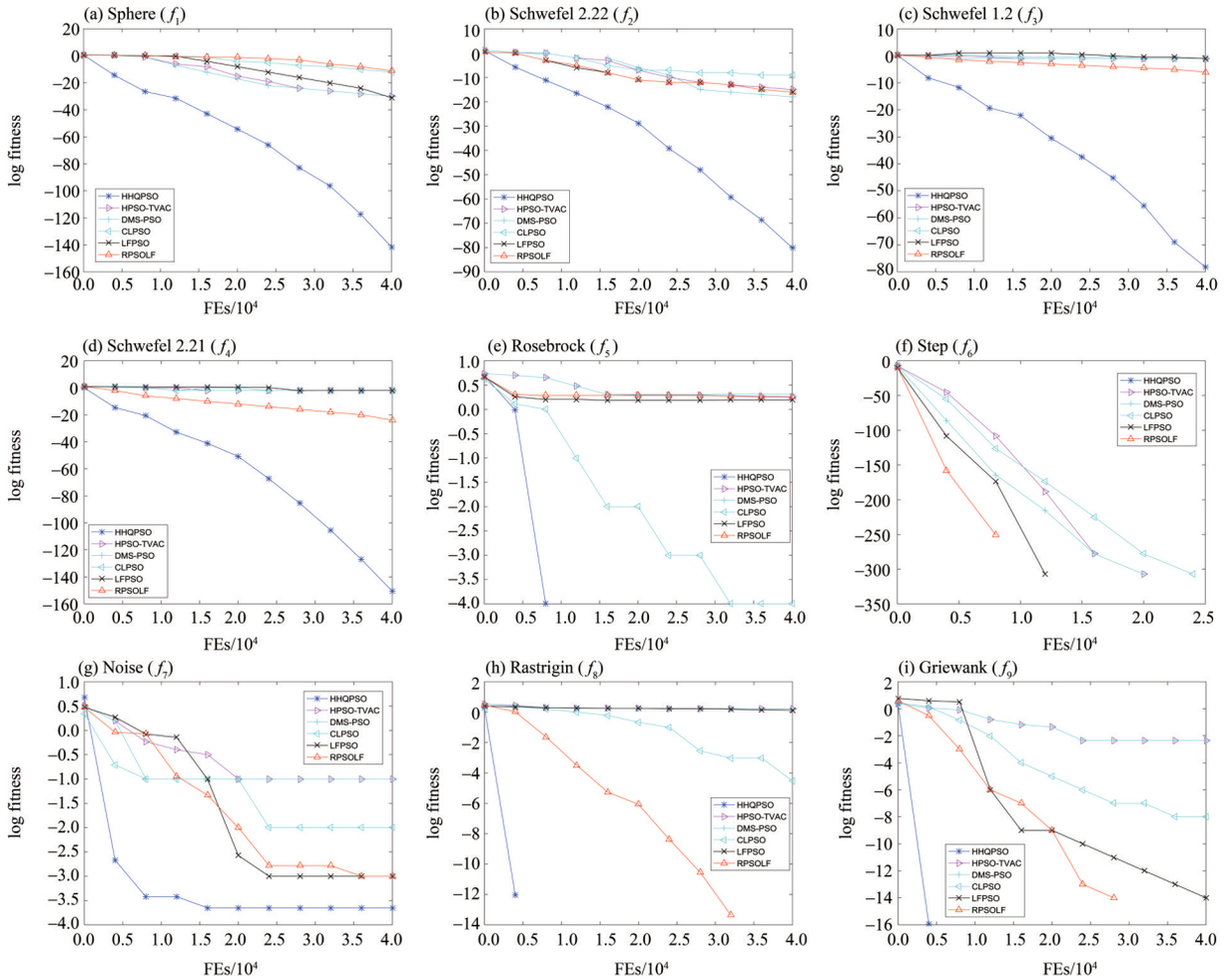

图2　各算法收敛效果对比

Fig. 2　Comparison of six algorithms on convergence

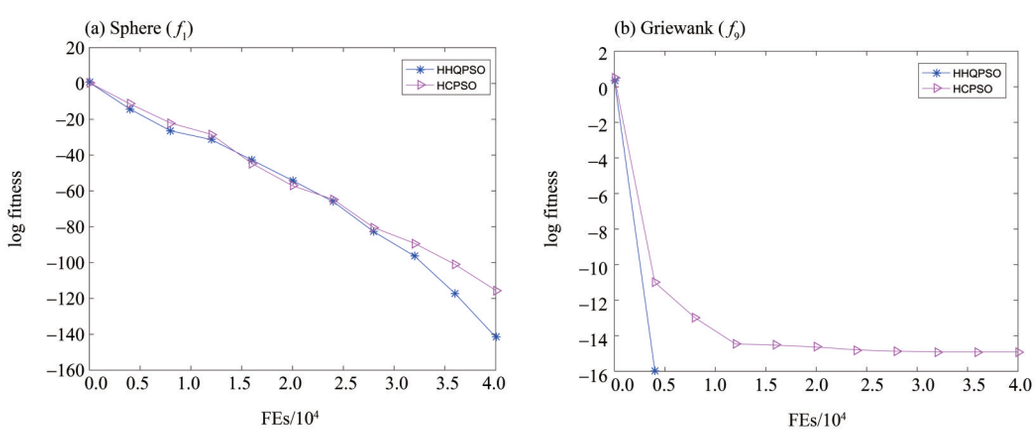

图3　HHQPSO与HCPSO收敛效果比较

Fig. 3　Comparison of HHQPSO and HCPSO on convergence

中下三层，在不同阶层的粒子最优位置都不一样，以此作为学习模型选择的依据，有针对性地去提高粒子所在区域的搜索精度。这两种方式的结合更好地开发种群的能力，同时还能够提高种群的勘探能力。然后对比基准测试函数和HHQPSO算法，从结果可以明显看到基准测试函数的稳定性、收敛速度和精度都要更好一些。

参考文献：

[1] 吴志攀，赵跃龙，罗中良，等．基于PSO-BP神经网络的车牌号码识别技术[J]．中山大学学报(自然科学版)，2017，56(1)：46-52.

[2] 陈洋波，徐会军，李计．流域洪水预报分布式模型参数自动优选[J]．中山大学学报(自然科学版)，2017，56(3)：125-133.

[3] 孙辉，邓志诚，赵嘉，等．优质个体最优动态空间变异的粒子群优化算法[J]．计算机应用研究，2020，37(8)：2344-2348+2370.

[4] 张鹏威．采用正弦映射与扩张算子的二进制粒子群优化算法[J]．小型微型计算机系统，2019，40(6)：1160-1164.

[5] CHEN K，ZHOU F，YIN L，et al. A hybrid particle swarm optimizer with sine cosine acceleration coefficients [J]. Information Sciences，2018，422：218-241.

[6] HAKLI H，UĞUZ H. A novel particle swarm optimization algorithm with Levy flight [J]. Applied Soft Computing，2014，23：333-345.

[7] LIANG J J，SUGANTHAN P N. Dynamic multi-swarm particle swarm optimizer [C]//Proceedings of 2005 IEEE Swarm Intelligence Symposium，2005：124-129.

[8] 袁小平，蒋硕．基于分层自主学习的改进粒子群优化算法 [J]．计算机应用，2019，39(1)：148-153.

[9] WANG D J，SUN J，XU W B. Quantum-behaved particle swarm optimization for security constrained economic dispatch [C]//The Fifth International Conference on Distributed Computing and Applications for Business Engineering and Sciences，Hangzhou，2006：42-46.

[10] 佃松宜，梁伟博，赵涛．基于改进QPSO的两轮移动机器人区间二型模糊逻辑控制 [J]．控制与决策，2019，34(2)：261-268.

[11] 丁青锋，尹晓宇．差分进化算法综述 [J]．智能系统学报，2017，12(4)：431-442.

[12] ZHANG H，XIE J，HU Q，et al. A hybrid DPSO with levy flight for scheduling MIMO radar tasks [J]. Applied Soft Computing，2018(71)：242-254.

[13] 曹天问，雷秀娟，杜明煜．一种基于Levy飞行的细菌觅食优化算法 [J]．计算机应用研究，2015，32(9)：2601-2605.

[14] RATNAWEERA A，HALGAMUGE S K，WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Transactions on Evolutionary Computation，2004，8(3)：240-255.

[15] LIANG J J，QIN A K，SUGANTHAN P N，et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation，2006，10(3)：281-295.

[16] YAN B，ZHAO Z，ZHOU Y，et al. A particle swarm optimization algorithm with random learning mechanism and Levy flight for optimization of atomic clusters [J]. Computer Physics Communications，2017，219：79-86.

[17] CHEN K，ZHOU F，YIN L，et al. A hybrid particle swarm optimizer with sine cosine acceleration coefficients [J]. Information Sciences，2018，422：218-241.

（责任编辑　冯兆永）