

层次学习骨干粒子群优化算法

申元霞^a, 陈健^a, 曾传华^b, 汪小燕^a, 卫琳娜^a

(安徽工业大学 a. 计算机科学与技术学院, b. 数理科学与工程学院, 安徽 马鞍山 243032)

摘要: 对骨干粒子群优化(BPSO)种群多样性迅速丧失的原因进行分析, 提出层次学习骨干粒子群优化算法以克服早熟现象. 该算法中粒子依不同的学习概率向粒子自身的最优粒子、优胜粒子和群体最优粒子学习, 该机制使群体实现不同层次的搜索并有效维持群体的多样性. 此外, 群体最优粒子依概率采用跳跃策略以增强逃逸能力或采用扰动策略以提高解的质量. 将所提出的算法与多种改进的粒子群优化算法进行对比, 仿真结果表明, 所提出算法的综合表现优于其他算法.

关键词: 骨干粒子群优化; 早熟; 种群多样性

中图分类号: TP301.6

文献标志码: A

Hierarchical learning bare-bones particle swarm optimization algorithm

SHEN Yuan-xia^a, CHEN Jian^a, ZENG Chuan-hua^b, WANG Xiao-yan^a, WEI Lin-na^a

(a. School of Computer Science and Technology, b. School of Mathematics & Physics, Anhui University of Technology, Maanshan 243032, China. Correspondent: SHEN Yuan-xia, E-mail: yuanxiashen@163.com)

Abstract: The reasons for the fast loss of population diversity in bare-bones particle swarm optimization(BPSO) are analyzed, and a hierarchical learning BPSO(HLBPSO) is developed to avoid the premature convergence. In the HLBPSO, each particle can learn from the personal best(pbest) particle, the superior particle and the global best(gbest) particle according to the learning probability. The learning mechanisms can provide hierarchical searching for maintaining the swarm diversity. Moreover, the gbest particle adopts the jump strategy to strengthen the escape ability and the disturbance strategy to improve the quality of the solutions. The experimental results show that the proposed method significantly outperforms the state-of-the-art PSO algorithms in terms of convergence speed and solution accuracy.

Keywords: BPSO; premature convergence; population diversity

0 引言

粒子群优化算法^[1](PSO)模拟鸟群、鱼群的觅食特性以及人类的社会行为, 由于其概念简单、易于实现, 受到了研究者的广泛关注. 2003年, Kennedy提出了骨干粒子群优化^[2](BPSO), 该算法消去了传统PSO中的速度项, 通过基于个体最优位置(pbest)和群体最优位置(gbest)的高斯采样完成粒子位置的更新. 与传统的PSO相比, BPSO减少了惯性系数、加速系数、速度阈值等控制参数, 使得算法结构更为简洁, 易于操作. 目前, BPSO在经济调度, 图像处理 and 故障诊断等实际问题中的应用取得了优良的效果^[3-4].

由于BPSO取消了速度项, 粒子直接学习pbest与gbest的算术平均值, 该学习机制虽然加快了算法的收敛速度, 但同时也增加了群体陷入早熟或局

部极值的机会. 为了改善BPSO早熟问题, 研究者提出了多种辅助策略以增强粒子的勘探能力, 如跳跃策略^[5-6]、突变策略^[7]、差分算子^[8]等. Zhang等^[9]提出了具有自适应的BPSO, 在该算法中对粒子更新位置的高斯采样的方差加入 Δ 增量, 且该增量受粒子的pbest和群体gbest的适应度控制; Liu等^[10]提出了融合反向学习的BPSO, 该算法将群体中所有的pbest和所有pbest的反向解按适应度值进行排序, 适应度高的解作为群体新的pbest; 张震等^[11]提出了剪枝策略的BPSO, 通过改进进化方程和剪枝策略保持种群多样性.

种群多样性的丧失是BPSO遭遇早熟的一个重要原因. 本文分析导致BPSO种群多样性丧失的因素, 讨论维持种群多样性的策略, 提出层次学习BPSO

收稿日期: 2015-12-22; **修回日期:** 2016-03-11.

基金项目: 国家自然科学基金项目(61300059, 61502010).

作者简介: 申元霞(1979-), 女, 副教授, 博士, 从事智能计算、智能信息处理等研究; 陈健(1993-), 男, 硕士生, 从事群体智能算法、图像处理的研究.

(HLBPSO)算法. 在算法中, 粒子依概率向粒子自身的 pbest、优胜粒子和 gbest 学习, 其中学习 pbest 利于勘探, 学习优胜粒子利于协调勘探与开采, 而学习 gbest 则注重解的开采. 群体实施不同层次的搜索可以有效维持群体的多样性. 同时, gbest 粒子依概率采用跳跃策略或扰动策略以增强粒子在进化前期的逃逸能力和进化后期的开采能力.

1 BPSO 算法

Kennedy 提出了骨干粒子群优化 (BPSO) 算法, 算法中没有速度, 粒子 i 的位置更新方程式如下:

$$\begin{aligned} x_{i,j}(t+1) &= \mu_{i,j}(t) + \sigma_{i,j}(t)N(0, 1), \\ \mu_{i,j}(t) &= 0.5(P_{i,j}(t) + G_j(t)), \\ \sigma_{i,j}(t) &= |P_{i,j}(t) - G_j(t)|. \end{aligned} \quad (1)$$

其中: $x_{i,j}(t+1)$ 是 $t+1$ 时刻粒子 i 位置的第 j 维分量, $i = 1, 2, \dots, N, j = 1, 2, \dots, D, N$ 是种群规模, D 是待求解问题的维数; P_i 是粒子 i 自身的最优位置 (i.e. pbest); G 是群体最优位置 (i.e. gbest); 符号 $N(0, 1)$ 是标准高斯分布. 由式 1 可知, 粒子的更新位置是以 pbest 与 gbest 的算术平均值 μ_{ij} 以及 pbest 与 gbest 的距离为标准差 σ_{ij} 并通过高斯采样得到的.

2 种群多样性分析

BPSO 搜索优化性能受种群多样性的影响, 高的种群多样性表明群体具有良好的勘探能力, 有助于逃离局部极值点, 从而减少早熟现象的产生, 但是过高的多样性会导致收敛速度缓慢. 低的种群多样性会加快群体的快速收敛, 但是在进化早期, 低的种群多样性则易于使群体陷入早期收敛. 合理地控制多样性可以平衡群体的勘探与开采. 下面将分析影响 BPSO 种群多样性的因素.

定义 1 基于种群重心的种群位置多样性定义如下^[12]:

$$I(X(t)) = \frac{1}{ND} \sum_{i=1}^N \sum_{d=1}^D [X_{id}(t) - \overline{X_d(t)}]^2. \quad (2)$$

其中: $\overline{X_d(t)} = (1/N) \sum_{i=1}^N X_{id}(t)$ 是群体所有粒子位置 $X(t)$ 重心的第 d 维分量. 当群体中的粒子位置 $X(t)$ 确定时, 由定义 1 可以计算出当前种群多样性.

令当前时刻为 t , 由式 1 可知, 粒子 $t+1$ 时刻的位置是随机的, 因此种群 $t+1$ 时刻的多样性也是随机的. 鉴于 $t+1$ 时刻种群多样性的随机性, 考察种群多样性的期望效用

$$\begin{aligned} E[I(X(t+1))] &= \\ \frac{1}{ND} \sum_{i=1}^N \sum_{d=1}^D E[X_{id}(t+1) - \overline{X_d(t+1)}]^2. \end{aligned} \quad (3)$$

定理 1 BPSO 在 $t+1$ 时刻的种群多样性期望为 $E[I(X(t+1))] = \frac{N-1}{N^2D} \sum_{i=1}^N \sum_{d=1}^D \sigma_{i,d}^2 + \frac{1}{2}I(P)$, 其中 $I(P)$ 是群体中 pbest 的多样性.

证明 由式 (3) 变形可得

$$\begin{aligned} E[I(x(t+1))] &= \\ \frac{1}{ND} \{E[X_{id}(t+1) - EX_{id}(t+1)]^2 - \\ 2E(E[X_{id}(t+1) - EX_{id}(t+1)][\overline{X_d(t+1)} - \\ X_{id}(t+1)]) + E[\overline{X_d(t+1)} - X_{id}(t+1)]^2\}. \end{aligned} \quad (4)$$

由式 (1) 可得, 随机变量 X_{id} 的方差为 σ_{ij}^2 , 期望为 μ_{ij} , 计算下式:

$$\begin{aligned} E([X_{id}(t+1) - EX_{id}(t+1)] \times \\ [\overline{X_d(t+1)} - EX_{id}(t+1)]) &= \\ 1/N \left(E \left([X_{id}(t+1) - EX_{id}(t+1)] \times \right. \right. \\ \left. \left. \left[\sum_{k=1}^N X_{kd}(t+1) - N \cdot EX_{id}(t+1) \right] \right) \right) &= \\ 1/N (E[X_{id}(t+1) - EX_{id}(t+1)]^2) &= (1/N) \sigma_{i,d}^2, \end{aligned} \quad (5)$$

$$\begin{aligned} E[\overline{X_d(t+1)} - X_{id}(t+1)]^2 &= \\ \frac{1}{N^2} \sum_{k=1}^N \sigma_{k,d}^2 + (E[\overline{X_d(t+1)} - X_{id}(t+1)])^2 &= \\ \frac{1}{N^2} \sum_{k=1}^N \sigma_{k,d}^2 + \left(\mu_{i,d} - \frac{1}{N} \sum_{k=1}^N EX_{k,d}(t+1) \right)^2 &= \\ \frac{1}{N^2} \sum_{k=1}^N \sigma_{k,d}^2 + \frac{1}{2} [P_{i,d} - \overline{P_d}]^2. \end{aligned} \quad (6)$$

将式 (5) 和 (6) 代入 (4), 可得

$$\begin{aligned} E[I(X(t+1))] &= \\ \frac{1}{ND} \sum_{i=1}^N \sum_{d=1}^D \left\{ \sigma_{i,d}^2 - \frac{2\sigma_{i,d}^2}{N} + \frac{1}{N^2} \sum_{k=1}^N \sigma_{k,d}^2 + \right. \\ \left. 0.5 [P_{i,d} - \overline{P_d}]^2 \right\} &= \\ \frac{N-1}{N^2D} \sum_{i=1}^N \sum_{d=1}^D \sigma_{i,d}^2 + \frac{1}{2} I(P), \end{aligned} \quad (7)$$

其中 $I(P)$ 是种群中 pbest 的多样性. \square

在进化初期群体没有聚集时, BPSO 中 pbest 与 gbest 保持一定的距离, 使得定理 1 中的方差 σ_{ij}^2 和 pbest 多样性 $I(P)$ 具有一定的值. 由于随着 pbest 不断向 gbest 逼近, 方差 σ_{ij}^2 不断减小; 同时群体向单一的 gbest 学习, 使得 pbest 之间也趋于相同, pbest 多样性 $I(P)$ 降低. 上述两方面的因素致使种群多样性快速下降. 另一方面, BPSO 中 gbest 是群体中最优秀的 pbest 个体, 但是该粒子采用式 (1) 进行更新时, 因

高斯采样方差为零, 丧失了学习能力. 由于 gbest 不具备逃逸能力, 当群体陷入局部极值时无法跳出.

由定理 1 可知, 种群规模与种群多样性的期望存在非线性关系, 单纯地增大种群规模未必利于多样性的保持. 该结论进一步证实了 Bergh 从实验中获得的结论, 即种群规模设置过大将降低优化性能^[13]. 因此, 群体维持多样性需合理地调控粒子高斯采样的方差和 pbest 差异性, 同时需增强 gbest 的逃逸能力.

3 HLBPSO

3.1 层次学习策略

为了保持种群中 pbest 的多样性, 提出层次学习策略, 即粒子依据学习概率向不同的导引粒子学习. 将引导粒子分为 3 个层次, 分别为粒子的 pbest、优胜榜样和群体的 gbest. 优胜榜样是指适应度优良的 pbest, 具体定义如下:

定义 2 优胜榜样. 设群体历史最优个体集合 $P = \{P_1, \dots, P_i, \dots, P_N\} (i \in 1, 2, \dots, N)$. 对于随机变量 $t, k (t, k \in 1, 2, \dots, N, t \neq k)$, 有 $F(P_k) > F(P_t)$, 则粒子 i 的第 j 维优质学习榜样是 P_{kj} .

粒子依固定的学习概率向优胜粒子学习, 利于协调勘探与开采; 随迭代次数的增加, 粒子依时变学习概率由进化初期加强 pbest 学习转向注重 gbest 学习, 该方式利于进化前阶段的勘探和后阶段的开采. 具体的粒子更新公式如下:

$$X_{i,j}(t+1) = \begin{cases} P_{i,j}^*(t) + |P_{i,j}^*(t) - P_{i,j}(t)| \times N(0, 1), & r_1 < 0.5; \\ X_{i,j}^*(t+1), & \text{otherwise.} \end{cases} \quad (8)$$

$$X_{i,j}(t+1) = \begin{cases} P_{i,j}(t) + |P_{r_3,j}(t) - P_{i,j}(t)| \times N(0, 1), & r_2 > pc; \\ 0.5(P_{i,j}(t) + G_j(t)) + |G_j(t) - P_{i,j}(t)| \times N(0, 1), & \text{otherwise.} \end{cases} \quad (9)$$

其中: $P_{i,j}^*(t)$ 是 $P_{i,j}(t)$ 的优胜榜样; r_1, r_2 是 $[0, 1]$ 分布的随机数; $P_{r_3,j}(t)$ 是种群中任意一个粒子的 pbest, $r_3 \in \{1, 2, \dots, N\}$; $pc = 0.5(1 - t/T_{\max})$ 是自适应学习概率, T_{\max} 是最大迭代次数, pc 随迭代次数 t 的增大而减小.

由式 (8) 和 (9) 可知, 粒子以 0.5 的概率学习优胜榜样, 以 $0.5pc$ 的概率学习 pbest, 以 $0.5(1-pc)$ 的概率学习 gbest 与 pbest 的均值. 随着迭代次数的增大, pc 的取值从 0.5 到 0 递减, 表明向 pbest 的学习概率逐渐递减, 同时向 gbest 与 pbest 的均值学习的概率逐渐增

大. 群体中的粒子向不同的引导粒子学习, 避免了向单一 gbest 学习而引起趋同化, 有效维持 pbest 的多样性. 另一方面, 粒子更新位置的高斯采样的标准差也是动态更新的, 延缓了 BPSO 中标准差快速下降的情况. 层次学习从上述两个方面的设计可以有效保持群体的多样性.

3.2 gbest 学习机制

在 HLBPSO 算法中, 对 gbest 实施了双策略学习, 即 gbest 依概率采用跳跃策略以增强逃逸能力, 采用扰动策略以增强开采能力.

gbest 跳跃策略的更新方程如下:

$$G_d^J(t+1) = \begin{cases} U(X_{d,\min}, X_{d,\max}), & \xi_1 < \theta; \\ G_d(t), & \text{otherwise.} \end{cases} \quad (10)$$

其中: ξ_1 是 $[0, 1]$ 均匀分布的随机数; $X_{d,\max}$ 和 $X_{d,\min}$ 分别是搜索空间在第 d 维的上界和下界, $U(X_{d,\min}, X_{d,\max})$ 符号表示在区间 $[X_{d,\min}, X_{d,\max}]$ 随机均匀取值; θ 是固定值. gbest 粒子的每维以概率 θ 进行跳跃, 使 gbest 摆脱当前位置, 本文中 θ 的设置参照文献 [6].

gbest 扰动策略更新方程如下:

$$G^*(t+1) = G(t) + \text{sig}(\xi_2) \times \xi_3 \times (G(t) - P_{\xi_4}). \quad (11)$$

其中: ξ_2 是在 $[-1, 1]$ 内均匀分布的随机数; ξ_3 是在 $[0, 1]$ 内均匀分布的随机数; ξ_4 是 $[1, N]$ 区间的随机整数, P_{ξ_4} 是群体中任意一个 pbest; $\text{sig}(\xi_2)$ 是符号函数, 如果 $\xi_2 \in [-1, 0]$, 则 $\text{sig}(\xi_2)$ 为负号, 否则 $\text{sig}(\xi_2)$ 为正号.

gbest 依 pc 的概率采用跳跃策略, 或者采用扰动策略

$$G_d'(t+1) = \begin{cases} G_d^J(t+1), & \xi_5 < pc; \\ G^*(t+1), & \text{otherwise.} \end{cases} \quad (12)$$

其中: pc 取值是随着迭代次数的变化由 1 递减到 0, gbest 在进化初期偏向采用跳跃策略而后期偏向扰动策略. 若 gbest 学习后的新粒子 G' 的适应度值高于当前的 gbest 适应度, 则更新, 否则当前 gbest 保持不变.

3.3 HLBPSO 算法流程

算法具体步骤如下:

Step 1: 初始化种群. 初始化种群位置 $X_i (i = 1, 2, \dots, N)$, 并将 X_i 赋值给 P_i .

Step 2: 计算种群位置 X_i 的适应度, 并将适应度最大的粒子的位置赋值给群体最优个体 G .

Step 3: 按式 (8) 和 (9) 产生粒子 X_i 的新位置, 计算当前粒子的适应度值, 依据适应度值更新 P_i 和 G .

Step 4: 按照式 (10)~(12) 产生 G' , 计算其适应度值, 如果 G' 的适应度大于 G 的适应度, 则将 G' 赋值给 G .

Step 5: 判断迭代次数是否达到最大迭代次数, 如

若满足条件, 则停止迭代, 将 G 作为结果输出, 否则转至 Step 3.

HLBPSO 算法流程图如图 1 所示.

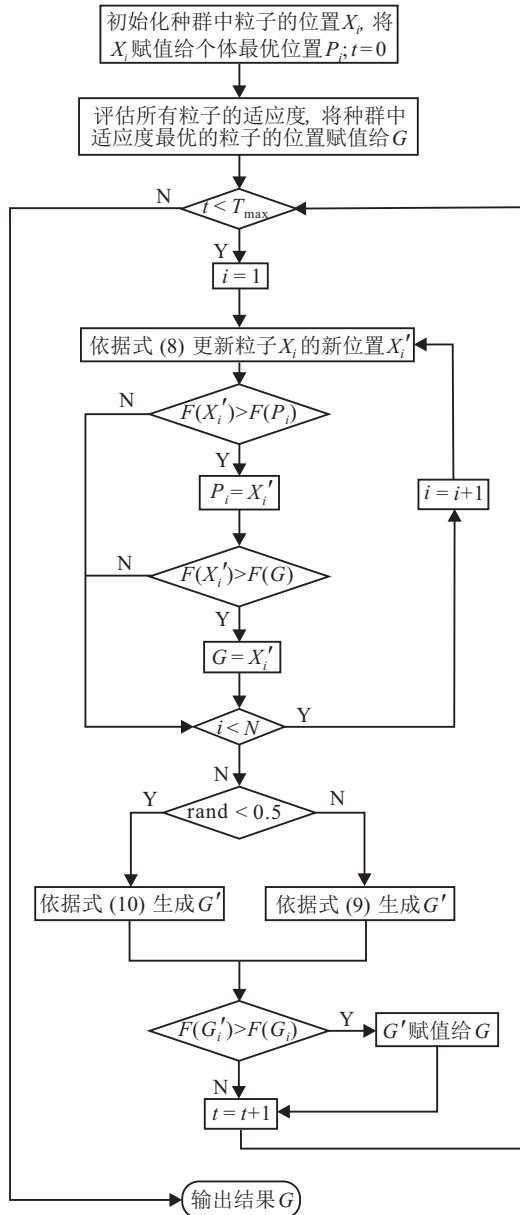


图 1 HLBPSO 算法流程

由流程图可知: Step 1 中粒子群体初始化的复杂度是 $O(ND)$, N 是规模, D 是搜索空间的维数; Step 2 中评价粒子适应度的复杂度是 $O(N)$; Step 3 和 Step 4 中粒子更新的复杂度是 $O(ND) + O(D)$. HLBPSO 的时间复杂度是 $O(ND) + O(D)$, BPSO 的计算复杂度是 $O(ND)$, HLBPSO 的计算开销与 BPSO 是在同一数量级上的.

3.4 HLBPSO 算法收敛性和复杂度分析

文献 [9] 给出了基于随机分析方法的 BPSO 的收敛性分析, 本文采用该方法对 HLBPSO 进行收敛性分析. 不失一般性, 将 HLBPSO 简化为一维模型, 并用下式表示:

$$X_i(t+1) = \begin{cases} \mu_i(t) + \sigma_i(t)N(0, 1), & r_1 > \eta; \\ X_i(t), & \text{otherwise.} \end{cases} \quad (13)$$

其中: η 为选择概率; $\mu(t)$ 和 $\sigma(t)$ 表示如下:

$$(\mu_i(t), \sigma_i(t)) = \begin{cases} (P_i^*, |P_i^* - P_i(t)|), & r_2 \leq 0.5; \\ (P_i, |P_{r3} - P_i(t)|), & 0.5 < r_2 \leq 0.5 + 0.5pc; \\ (0.5(P_i(t) + G(t)), |G(t) - P_i(t)|), & 0.5 + 0.5pc < r_2 \leq 1. \end{cases} \quad (14)$$

其中 r_1 和 r_2 均为 $[0,1]$ 区间的随机数.

定理 2 当且仅当 $0 < \eta \leq 1$ 时, 迭代序列 $X_i(t)$ 的均值 $(E(X_i(t)))$ 收敛于 $0.5(P_i(t) + G(t))$.

证明 依据式 (13) 和 (14) 可以得到 $E(X_i(t))$ 的迭代方程如下:

$$E(X_i(t+1)) = (1 - \eta)E(X_i(t)) + \eta(0.5P_i^*(t) + 0.5pcP_i(t) + 0.25(1 - pc)(P_i(t) + G(t))). \quad (15)$$

式 (15) 的特征方程为 $\lambda + 1 - \eta = 0$.

$(E(X_i(t)))$ 迭代收敛的条件为 $\lambda < 1$, 只要 $0 < \eta \leq 1$, 即可满足条件. 随着迭代次数的增加, 粒子学习 P_i 与 G 中点的概率增大, 致使优胜粒子 P_i^* 不断靠近 $0.5(P_i + G)$, 同时 pc 值也趋近 0. 由式 (15) 可得

$$\lim_{t \rightarrow \infty} E(X_i(t)) = \lim_{\substack{P_i^* \rightarrow 0.5(P_i + G) \\ pc \rightarrow 0}} (0.5P_i^* + 0.5pcP_i + 0.25(1 - pc)(P_i + G)) = \frac{P_i + G}{2}. \quad \square$$

4 数值实验及结果分析

4.1 测试函数和参数设置

为了验证本文算法的优化性能, 采用两组测试函数: 第 1 组为经典多峰函数, 如表 1 所示; 第 2 组为带旋转漂移函数 (CEC2005)^[14]. 所有函数的维度为 30. 对第 1 组函数, 本文算法与经典 BPSO^[2]、BPSODE^[8]、BPSOMC^[7]、BPSO_J^[6] 和 ABPSO^[9] 进行实验比较, 对比算法的参数设置均参照相应文献. 其中: BPSODE 的重组概率是 0.7; BPSOMC 的突变概率是 0.5, 方差固定值是 0.001, 领域规模是 2; BPSO_J 的学习率是 0.7, 进化代数是 30000 代, 运行次数是 30 次, 种群规模是 20. 对于第 2 组函数, 本文算法不仅与经典 BPSO^[2] 算法比较, 还与知名 PSO 算法 CLPSO^[15] 和 SRPSO^[16] 进行比较, 其中 BPSO 和 PSO 改进算法的结果均来自文献 [16]. 为了保证比较的公正性, 本文算法的进化代数、运行次数、种群规模与文献 [16] 相同. 本文采用适应度成功率、均值和方差评价算法的优化性能, 其中适应度值是 $(f(x) - f(x^*))$ (x^* 和 x 分别是理论最优值和算法所获得的最优值), 当算法达

到 10^{-8} 时, 即认为算法成功找到最优值.

表 1 第 1 组测试函数

函数	函数名	搜索空间	最优值
f_1	Griewank	$[-600, 600]$	0
f_2	Ackley	$[-32, 32]$	0
f_3	Rastrigin	$[-5.12, 5.12]$	0
f_4	Schewfel 2.26	$[-500, 500]$	-12.569.5
f_5	Weierstrass	$[-0.5, 0.5]$	0

4.2 第 1 组数值实验结果分析

第 1 组测试函数的结果如表 2 所示, 粗体显示了获得的最优结果.

表 2 7 种算法在 5 个测试函数上的均值、方差和成功率

函数		BPSO	BPSOMC	BPSODE	BPSO-J	ABPSO	HLBPSO
f_1	FM(SD)	1.23e-02(7.85e-03)	0.00e+00(0.00e+00)	3.18e+00(1.46e+00)	2.11e-01(1.76e-02)	1.86e-12(8.66e-13)	0.00e+00(0.00e+00)
	SR/%	40	100	0	0	100	100
f_2	FM(SD)	7.11e-15(0.00e+00)	3.20e-14(6.51e-15)	6.85e+00(4.16e-01)	2.62e-03(1.54e-03)	1.46e-07(4.01e-08)	4.33e-15(1.05e-15)
	SR/%	100	100	0	70	0	100
f_3	FM(SD)	4.28e+01(1.32e+01)	1.19e+01(9.04e-01)	2.88e+01(3.61e+00)	8.63e-03(4.31e-03)	2.98e+00(1.02e+00)	0.00e+00(0.00e+00)
	SR/%	0	0	0	0	0	100
f_4	FM(SD)	1.95e+03(6.44e+02)	8.09e+02(2.51e+02)	1.99e+03(3.13e+02)	1.50e-02(1.38e-02)	6.49e-09(1.27e-09)	0.00e+00(0.00e+00)
	SR/%	0	0	0	0	100	100
f_5	FM(SD)	1.54e+00(2.63e-01)	9.04e-14(1.28e-14)	4.10e+00(3.10e-01)	1.32e-01(6.09e-02)	6.61e-04(8.11e-05)	0.00e+00(0.00e+00)
	SR/%	0	100	0	70	100	100

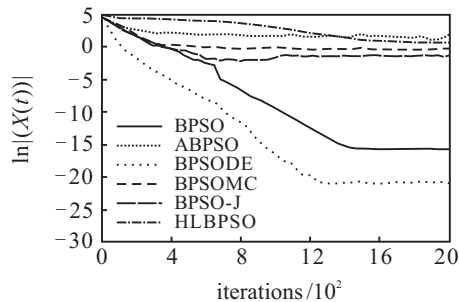


图 2 Griewank 函数多样性变化

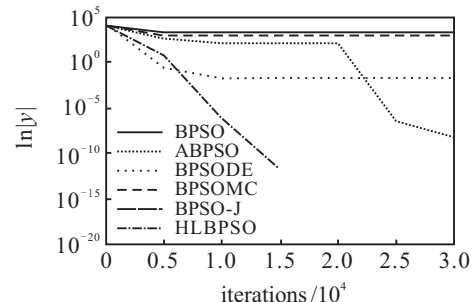


图 5 Schewfel 2.26 进化

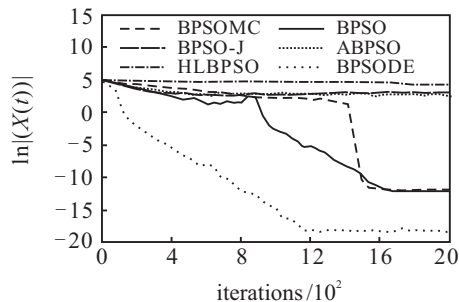


图 3 Schewfel 2.26 函数多样性变化

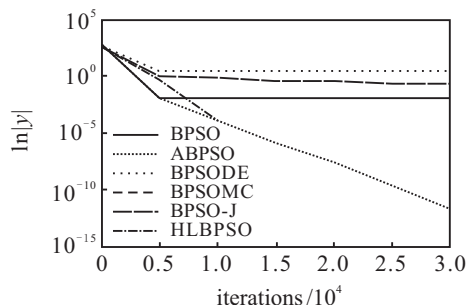


图 4 Griewank 函数进化

各个算法在 Griewank 和 Schewfel 2.26 函数上的多样性的变化曲线如图 2 和图 3 所示, 各个算法在 Griewank 和 Schewfel 2.26 函数上的进化曲线如图 4 和图 5 所示.

为了便于比较, 图 2~图 5 的纵坐标为算法获取多样性和最优值的对数值. 对于多模函数 $f_1 \sim f_5$, HLBPSO 找到了 f_1 、 f_3 、 f_4 和 f_5 理论最优解, 对于函数 f_2 也获取了较高的收敛精度. 从寻优的成功率上可知, HLBPSO 寻优成功率均是 100%, 而 BPSO 只在 Ackley 函数上获得了 100% 的成功率.

由图 2 和图 3 可以看出: BPSO 和 BPSODE 算法的多样性快速下降, 而 HLBPSO 算法在进化前期保持了较高的多样性, 说明层次学习策略利于维持多样性. 从进化曲线可知: BPSO 和 BPSODE 算法陷入早期收敛, 说明低的多样性将易导致种群陷入局部最优. 在进化后期, 高的多样性不利于精细搜索. ABPSO 算法一直保持较高的多样性, 由进化曲线可知, 该算法在进化后期仍具有较强的探测能力, 但是延缓了收敛速度. HLBPSO 的多样性随着迭代次数的增加呈现下降趋势, 使得 HLBPSO 在上述两个函数上不仅找到全局最优解, 而且均在 2 万代内收敛, 较好地平衡了算法的精度和速度.

4.3 第 2 组数值实验结果分析

第 2 组测试函数的结果如表 3 所示, 粗体显示了所获得的最优结果.

在表 3 中: PSO、BPSO、CLPSO 和 SRPSO 算法的实验结果均来自文献 [14], 文献 [14] 表明, SRPSO

和 CLPSO 具有良好的优化性能.

由表 3 可知, HLBPSO 在 8 个函数上获得了最优的结果, 并找到了 F_1 和 F_9 的理论最优值. 为了综合

比较算法之间的性能, 本文采用 Friedman 检验对结果进行分析, 表 4 给出了 5 种算法的平均排名, 本文算法综合性能第 1.

表 3 算法在 CEC2005 测试函数上的均值

算法	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
λ PSO	9.65e+00	1.57e+01	1.02e+07	1.83e+03	8.10e+03	1.17e+03	6.78e+03	2.09e+01	6.54e+01	8.66e+01	2.79e+01	1.87e+04
BPSO	0.00e+00	9.26e-03	1.29e+06	2.30e+03	5.31e+03	2.79e+01	4.69e+03	2.09e+01	5.63e+01	7.55e+01	2.80e+01	2.58e+03
CLPSO	0.00e+00	3.82e+02	1.18e+07	5.39e+03	4.00e+03	1.77e+01	4.69e+03	2.07e+01	0.00e+00	8.02e+01	2.52e+01	1.32e+04
SRPSO	0.00e+00	0.00e+00	4.51e+01	2.98e+01	1.75e+03	3.97e+01	7.61e-02	2.09e+01	3.75e+00	5.19e+01	9.58e+00	2.49e+03
HLBPSO	0.00e+00	5.68e-14	7.66e+05	8.52e-14	1.67e+03	1.41e+01	2.84e-14	2.06e+01	0.00e+00	7.47e+01	2.72e+01	1.80e+03

表 4 5 种算法的平均排名

算法	平均排名
λ PSO	4.5(5)
BPSO	3.5(4)
CLPSO	3(3)
SRPSO	2.1(2)
HLBPSO	1.3(1)

5 结 论

本文首先给出了 BPSO 下一时刻群体多样性的期望值, 并分析了引起种群多样性快速下降的因素; 然后, 提出了层次学习 BPSO (HLBPSO) 算法, 粒子依概率学习不同层次的引导粒子维持群体的多样性, 同时 gbest 依概率采用跳跃策略以增强逃逸能力或采用扰动策略以提高解的质量; 最后, 对 17 个标准函数进行测试, 统计结果验证了本文算法的有效性.

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of the IEEE Int Conf on Neural Networks. Perth: IEEE, 1995: 1942-1948.
- [2] Kennedy J. Bare bones particle swarms[C]. Proc of the IEEE Swarm Intelligence Symposium. Indianapolis: IEEE, 2003: 80-87.
- [3] Campos M, Krohling R A. Entropy-based bare bones particle swarm for dynamic constrained optimization[J]. Knowledge-Based Systems, 2016(97): 203-223.
- [4] Zhang Y, Gong D W, Hu Y, et al. Feature selection algorithm based on bare-bones particle swarm optimization[J]. Neurocomputing, 2015, 148(19): 150-157.
- [5] Krohling R A, Mendel E. Bare bones particle swarm optimization with Gaussian or Cauchy jumps[C]. Proc of the IEEE Congress on Evolutionary Computation. Trondheim: IEEE, 2009: 3285-3291.
- [6] Blackwell T. A study of collapse in bare bones particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2012, 16(3): 354-375.
- [7] Zhang H, Kennedy D D, Rangaiah G P, et al. A novel bare-bones particle swarm optimization and its performance for modeling vapor-liquid equilibrium data[J]. Fluid Phase Equilibria, 2011, 301(1): 33-45.
- [8] Omran M G H, Engelbrecht A P, Ayed S. Bare bones differential evolution[J]. European J of Operational Research, 2009, 196(1): 128-139.
- [9] Zhan Z H, Zhang J, Li Y, et al. Adaptive particle swarm optimization[J]. IEEE Trans on Systems, Man and Cybernetics—Part B: Cybernetics, 2009, 39(6): 1362-1381.
- [10] Liu H, Xu G, Ding G Y, et al. Integrating opposition-based learning into the evolution equation of bare-bones particle swarm optimization[J]. Soft Comput, 2015, 19(10): 2813-2836.
- [11] 张震, 潘再平, 潘晓弘. 基于剪枝策略的骨干粒子群算法[J]. 控制与决策, 2015, 30(9): 1591-1596. (Zhang Z, Pan Z P, Pan X H. Pruning strategy based bare bones particle swarm optimization[J]. Control and Decision, 2015, 30(9): 1591-1596.)
- [12] Shi Y H, Eberhart R C. Population diversity of particle swarms[C]. Proc of IEEE Congress Evolutionary Computation. HongKong: IEEE Press, 2008: 1063-1067.
- [13] Bergh F Van Den, Engelbrecht A P. Effect of swarm size on cooperative particle swarm optimizers[C]. Proc of Genetic Evolutionary Computation. San Francisco: IEEE Press, 2001: 892-899.
- [14] Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer[C]. Proc of the IEEE Int Conf on Swarm Intelligence Symposium. California: IEEE Press, 2005: 210-224.
- [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 Trans on Evol Comput, 2006, 10(3): 281-295.
- [16] Tanweer M R, Suresh S, Sundararajan N. Self regulating particle swarm optimization algorithm[J]. Information Sciences, 2015, 294(2): 182-202.