

## 基于剪枝策略的骨干粒子群算法

张震, 潘再平, 潘晓弘

(浙江大学 工学部, 杭州 310000)

**摘要:** 为了优化算法的全局探索能力和局部开发能力, 提出一种基于两方面改进的骨干粒子群算法. 提出一种进化方程, 通过即时搜索域的分析说明该方程可以改善粒子多样性. 提出粒子群“剪枝”策略: 每当粒子搜索到新的群体最优位置时, 剪去该粒子, 同时初始化一个新位置以安插该粒子. 理论分析指出, 在增强全局探索能力的同时, 合适的剪枝策略能增加局部开发能力. 实验结果表明, 所提出算法的性能较几种经典 PSO 算法有显著的提升.

**关键词:** 骨干粒子群; 剪枝策略; 粒子多样性; 全局探索能力; 局部开发能力

中图分类号: TP301.6

文献标志码: A

## Pruning strategy based bare bones particle swarm optimization

ZHANG Zhen, PAN Zai-ping, PAN Xiao-hong

(College of Engineering, Zhejiang University, Hangzhou 310000, China. Correspondent: PAN Zai-ping, E-mail: panzaiping@zju.edu.cn)

**Abstract:** A bare bones particle swarm optimization(NPSO) algorithm is proposed to improve both global exploration and local exploitation. An evolution equation which obtains better swarm diversity is employed in the NPSO algorithm. Inspired by the apical dominance phenomenon in biology, a particle pruning strategy is introduced as follows: When a particle reaches a new best position of the swarm, it would be pruned and inserted to another position. Theoretical analysis shows that the pruning strategy can improve both global exploration and local exploitation. Finally, results of the experiments on benchmark problems show that the proposed algorithm obtains significant improvement when compared to some classical PSO algorithms.

**Keywords:** bare bones PSO; particle pruning strategy; swarm diversity; global exploration; local exploitation

### 0 引言

骨干粒子群(BBPSO)是一种取消了粒子速度属性的粒子群算法, 以随机分布的形式完成粒子位置的进化. 与粒子群优化算法(PSO)一样, BBPSO具有良好的收敛性, 以及对问题的连续性和可导性无要求等优点. 此外, 由于去除了速度属性, BBPSO算法形式比 PSO 算法形式更加简洁, 理论结构更加简单. Kennedy<sup>[1]</sup>提出的高斯骨干粒子群和 Sun 等<sup>[2]</sup>提出的量子粒子群是研究最为广泛的两种 BBPSO 形式. 以此为基础, 近年来诞生了大量 BBPSO 的理论分析<sup>[3]</sup>、改进<sup>[4-5]</sup>和应用<sup>[6-7]</sup>等方面的研究.

早熟问题和“burst”现象是存在于各种 PSO 算法中的普遍问题: 早熟指粒子收敛于非全局最优解的位置, Van<sup>[8]</sup>证明了经典 PSO 不仅不能保证收敛于全局最优解, 而且不能保证收敛于局部最优解, 这一结论说明了早熟是必然存在的; burst 现象指粒子群在一段时间内几乎不进行进化, 而在一定迭代次数后突然又

恢复进化的现象, 这一过程降低了算法的鲁棒性.

解决早熟和 burst 问题, 改善算法性能, 需从优化算法全局探索能力和局部开发能力两方面入手, 前者代表粒子跳出当前局部最优的能力, 后者代表在当前邻域内细化搜索的能力. 以此为目标, 目前广泛研究的改进策略包括邻域控制(动态种群)、直接位置调节和控制参数调节. 刘衍民等<sup>[9]</sup>以聚类分析方法将种群分成若干子种群, 范成礼等<sup>[10]</sup>引入带审敛因子的变邻域方法, Blackwell 等<sup>[11]</sup>在 BBPSO 群体发生早熟时引入突变机制, 张顶学等<sup>[12]</sup>引入自适应参数调节措施.

本文提出一种基于剪枝策略的骨干粒子群算法(NPSO), 算法包括两方面改进: 1) 提出一种 BBPSO 进化方程, 该方程通过优化概率分布改善粒子多样性; 2) 引入粒子群“剪枝”策略. “剪枝”概念来源于粒子群早熟与生物学“顶端优势”现象的类比: 群体中最优粒子对种群的作用类似于植物“顶芽”对植物生长的影响, 发育最好的粒子(植物顶芽)一方面引导进

收稿日期: 2014-06-09; 修回日期: 2014-08-26.

作者简介: 张震(1986—), 男, 博士, 从事工程优化、进化计算的研究; 潘再平(1957—), 男, 教授, 从事电机系统与控制、新能源等研究.

化,另一方面使群体过于依赖当前搜索经验,从而产生早熟现象(顶端优势现象),影响粒子群进化(植物总体发育).受此启发提出一种粒子群“剪枝”策略:每当一个粒子搜索到新的群体最优位置时,剪去该粒子,同时初始化一个新位置以安插该粒子.

剪枝策略实施简易,产生的额外计算量小.本文从理论上证明了剪枝策略不仅能优化全局探索能力,而且能通过构造合适的剪枝策略提高局部开发能力.实验表明,NPSO算法在解决早熟和burst问题的同时,显著提升了算法的性能.

## 1 骨干粒子群算法简介

骨干粒子群的一般模型可以归纳如下:

$$X_i = \mu(L_i) + \alpha\xi\delta(L_i). \quad (1)$$

其中:矢量 $X_i$ 为粒子 $i$ 的进化位置; $\alpha$ 为标量,是唯一的控制参数, $\alpha$ 越大,粒子群越倾向于分散, $\alpha$ 越小,越倾向于聚拢; $L_i$ 为集合,表示邻域拓扑选择;矢量 $\mu$ 和 $\delta$ 均为 $L_i$ 的函数,分别表示进化中心项和离散控制项; $\xi$ 为标准随机变量.常用的控制方法包括固定策略、时变策略和自适应策略3类:自适应策略性能最好,但实施复杂,额外计算开销大;时变策略额外开销小、性能提升较好,是应用最广泛的方式.对于局部环状拓扑,有 $L_i = \{P_{i-1}, P_i, P_{i+1}\}$ ;对于全局邻域拓扑,有 $L_i = \{P_1, P_2, \dots, P_N\}$ ;对于动态邻域, $L_i$ 是时变的.其中: $P_i$ 为粒子 $i$ 的历史最优位置, $N$ 为粒子数.在BBPSO中, $\mu$ 为概率分布的期望位置,随着迭代的进行,粒子群将收敛于 $\mu$ . $\delta$ 控制搜索域中粒子分布方差的大小,一般情况下,在迭代后期, $\delta$ 将减小到0,从而使粒子群自然收敛于 $\mu$ .广泛使用的分布类型包括标准高斯分布、柯西分布、指数分布和Levy分布等. $\xi$ 可以取标量或对角矩阵两种形式,分别代表算法的两种实现:若 $\xi$ 为标量,则粒子的各个维度在一次进化中使用同一个随机变量,这是一种各向同性的实现;若 $\xi$ 为对角矩阵,即 $\xi = \text{diag}[\xi_1, \xi_2, \dots, \xi_D]$ ,则粒子各个维度使用独立的随机变量,这是一种各向异性的实现.对角矩阵实现的粒子多样性更优,求解多峰问题性能更好.

目前,高斯骨干粒子群(GBBPSO)和量子粒子群(QPSO)是两种影响最大的BBPSO形式.在GBBPSO中: $\alpha$ 固定为1;进化中心项 $\mu$ 为粒子 $i$ 历史最优位置 $P_i$ 与群体最优位置 $G$ 的中点;离散控制项 $\delta$ 为 $P_i$ 与 $G$ 的偏差; $\xi$ 为标准高斯分布 $N(0, 1)$ ,且有

$$X_i = \frac{P_i + G}{2} + \frac{\xi(P_i - G)}{2}. \quad (2)$$

在QPSO中: $\mu$ 为 $P_i$ 与 $G$ 中间的随机一点; $\delta$ 为所有粒子历史最优位置的几何中心与粒子 $i$ 当前位置的偏差; $\xi$ 为双指数分布,且有

$$X_i = \frac{\varphi_1 P_i + \varphi_2 G}{\varphi_1 + \varphi_2} + \alpha\xi\left(\frac{1}{N}\sum P_i - X_i\right). \quad (3)$$

其中: $\varphi_1$ 和 $\varphi_2$ 为 $[0, 1]$ 区间上均匀分布的变量,双指

数随机变量 $\xi$ 的概率密度函数为

$$f(x) = \begin{cases} \frac{1}{2}e^{-x}, & x > 0; \\ \frac{1}{2}e^x, & x \leq 0. \end{cases} \quad (4)$$

## 2 NPSO 算法

### 2.1 进化方程

本文采用如下骨干粒子群进化方程:

$$\mu(L_i) = \frac{\xi_a P_i + \xi_b G}{\xi_a + \xi_b}, \quad (5)$$

$$X_i = \mu(L_i) + \alpha\xi(\mu(L_i) - X_i). \quad (6)$$

其中: $\xi$ 、 $\xi_a$ 、 $\xi_b$ 均为标准高斯分布, $P_i$ 为粒子 $i$ 的历史最优位置, $G$ 为群体最优位置.

在BBPSO中,离散控制项有两种形式:1)粒子 $i$ 下一时刻的进化位置仅由 $L_i$ 决定,如GBBPSO;2)除了 $L_i$ ,粒子 $i$ 的进化位置还受当前位置 $X_i$ 的影响,如QPSO.在驻态时,即粒子的最优位置保持不变的状态下,显然形式1)的搜索域更加集中,形式2)更加发散.对于单峰函数,形式1)可能更有利,但是对于复杂多峰函数,实现2)将更易跳出局部最优.Sun等<sup>[2]</sup>通过实验指出,QPSO的总体性能优于GBBPSO,因此NPSO采用了形式2).

在标准PSO中,粒子 $i$ 将收敛于线段 $\overline{P_i G}$ 上一点.基于此,GBBPSO将分布中心 $\mu$ 固定为 $\overline{P_i G}$ 的中点,QPSO则将 $\mu$ 设为均匀分布于 $\overline{P_i G}$ 上的随机变量.本文将高斯分布引入 $\mu$ ,通过分析即时搜索域说明 $\mu$ 的引入为NPSO带来的特性.不失一般性,在二维空间中讨论NPSO的即时搜索域,分析中略去下标 $i$ ,NPSO的即时搜索域如图1所示.

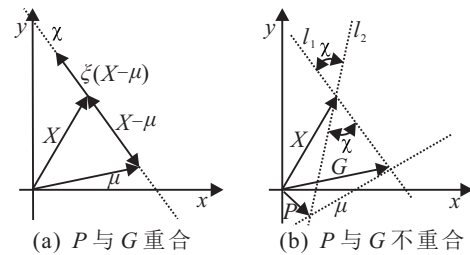


图1 NPSO的即时搜索域

在图1中:当 $P$ 与 $G$ 重合时, $\mu$ 将塌缩到一个点,此时的即时搜索域 $\chi$ 为一条直线,如虚线所示.可以看出,若 $\xi$ 为 $[0, 1]$ 均匀分布,则搜索域将塌缩到线段 $\overline{\mu X}$ 上.采用标准正态分布后,搜索域扩展到整条直线,由高斯分布概率密度公式可得,下一时刻粒子到达的位置 $X_i^{t+1}$ 出现在线段 $\overline{\mu X}$ 上的概率为

$$X_i^{t+1} = P\{X_i^{t+1} \in \{\alpha X_i - (1 - \alpha)L_i | 0 \leq \alpha \leq 1\}\} = P\{-1 \leq \xi \leq 1\} = 68.3\%. \quad (7)$$

在图1中:当 $P$ 与 $G$ 不重合时,分析同上,若 $\xi_a$ 和 $\xi_b$ 为 $[0, 1]$ 均匀分布,则搜索域 $\chi$ 为 $l_1$ 与 $l_2$ 之间的区域.采用标准正态分布后,搜索域扩展到整个空间,在 $l_1$ 与 $l_2$ 之间搜索的概率为68.3%.

综上所述, 在进化方程中引入了3个标准高斯分布变量  $\xi$ ,  $\xi_a$  和  $\xi_b$  之后, 粒子多样性得到了改善: 在着重搜索重点区域的同时, 以一定概率发散粒子, 使得算法不易早熟. 应说明的是, 在分析中, 假设  $\xi$  为标量, 即粒子进化时各个维度使用同一个  $\xi$ . 实际上, 无论  $\xi$  实现形式为对角矩阵还是标量, 对粒子多样性的改善都是相似的.

### 2.2 剪枝策略

在群体早熟时, 通过引入突变机制使粒子以一定方式离散是一种常用的策略. 这种思想最早起源于 Van<sup>[8]</sup> 的多次重启 PSO: 当粒子群汇聚于一点不再进化时, 人工重新初始化每个粒子的位置. 合适的突变机制可以使 PSO 成为全局收敛的算法: 当迭代次数趋向无穷时, 寻找到全局最优解的概率为 100%.

设当前最优位置离问题全局最优位置的距离为  $\delta_1$ , 粒子间的位置均方差为  $\delta_2$ , 若  $\delta_2$  比  $\delta_1$  更早下降到 0, 则发生求解早熟. 因而, 防止粒子群向最优粒子过早汇聚可以从本质上解决早熟问题. 受生物学中消除顶端优势的剪枝方法的启发, 本文提出一种突变机制——粒子群剪枝: 每当一个粒子  $i$  寻找到新的全局最优点  $P_i$  之后, 更新群体最优位置  $G = P_i$ , 寻找一个新位置作为粒子  $i$  的位置  $X'_i$  和历史最优位置  $P'_i$ , 具体初始化方法为

$$X'_{id} = P'_{id} = N(P_{id}, \delta^2). \quad (8)$$

其中:  $d$  为维度下标; 参数  $\delta$  为插枝方差, 以  $P_i$  为中心, 标准差为  $\delta$  的高斯分布重新设置每个维度的值.

与 Blackwell 等采取的突变机制不同, 剪枝策略并非在早熟时使用, 而是在每一次更新全局最优位置  $G$  时使用. 因此, 无需监视粒子群状态以判断早熟, 额外增加的计算量几乎可忽略.

与其他突变机制一样, 剪枝策略对全局探索能力的改善是显然的, 趋向于早熟时, 粒子更易于跳出当前局部最优点. 实际上, 剪枝策略同样能提升局部开发能力, 下面从理论上讨论若要改善局部开发能力, 剪枝策略需满足的条件.

将算法简化为 SAS (Somewhat adaptive search) 模型并进行分析. SAS 是由 Zabinsky 等<sup>[13]</sup> 提出的一种理想化的随机算法模型, 随着迭代的进行, 以固定的概率  $\rho$  ( $0 < \rho \leq 1$ ) 优化目标函数值.

**定义 1** 设问题的全局最优解为  $x^*$ , 对任意值  $\varepsilon$ , 定义  $K(\varepsilon)$  为算法第 1 次寻找到集合  $\{x | f(x) - f(x^*) \leq \varepsilon\}$  中的解时所耗费的迭代次数.

Sun 等<sup>[14]</sup> 证明了 SAS 模型中  $K(\varepsilon)$  的期望值为

$$E[K(\varepsilon)] = \frac{N}{\rho} \ln \left[ \frac{x}{\sqrt{\pi\varepsilon}} \right] + \ln \left[ \Gamma \left( \frac{N}{2} + 1 \right) \right]. \quad (9)$$

其中:  $N$  为问题的维数,  $x$  为优化范围.

不失一般性, 在一维空间下说明剪枝策略对求解过程的影响. 设最优解位于原点, 粒子当前位置为  $P$ , 剪枝策略可产生的两种情况如图 2 所示. 由图 2 可

得两类结果: 情况 1, 方差  $\delta$  较小, 从而在绝大部分情况下, 新位置与  $P$  位于原点的同一侧, 如在区间  $\overline{P_1P_2}$  内; 情况 2, 方差较大, 使粒子不再位于原点同一侧, 如主要分布于区间  $\overline{P_3P_4}$  内.

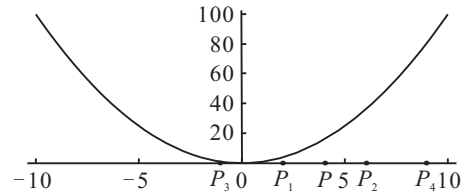


图 2 剪枝策略产生的两种情况

令  $a = \frac{N}{\rho}$ ,  $b = \ln \left[ \Gamma \left( \frac{N}{2} + 1 \right) \right] - \frac{N}{\rho} \ln \sqrt{\pi\varepsilon}$ , 若粒子从坐标  $p$  开始进化, 则到达精度  $\varepsilon$  所需的迭代次数期望为

$$E_1[K(\varepsilon)] = a \ln p + b. \quad (10)$$

对于情况 1,  $P_1$  与  $P$  的距离  $d$  小于  $P$  点坐标  $p$ , 则  $\overline{P_1P_2}$  内粒子  $K(\varepsilon)$  的平均期望值为

$$E_2[K(\varepsilon)] = \frac{1}{2d} \int_{p-d}^{p+d} (a \ln x + b) dx. \quad (11)$$

**定理 1** 若变量  $p$ ,  $d$  和常数  $a$ ,  $b$  满足  $p > d > 0$ ,  $a > 0$ , 则有如下关系成立:

$$\frac{1}{2d} \int_{p-d}^{p+d} (a \ln x + b) dx < a \ln p + b. \quad (12)$$

**证明** 记  $g(x) = x \ln x$ , 简化上式后, 只需证明

$$g(p+d) - g(p-d) < 2d(\ln p + 1). \quad (13)$$

构造如下函数:

$$f(p) = g(p+d) - g(p-d) - 2d(\ln p + 1). \quad (14)$$

由  $p > d > 0$  易知

$$f'(p) = \ln \frac{p+d}{p-d} - \frac{2d}{p} > 0, \quad (15)$$

因而  $f(p)$  单调上升. 令  $x = \frac{1}{p}$ , 并对  $f(p)$  求极限

$$\lim_{p \rightarrow +\infty} f(p) = \lim_{x \rightarrow 0} \left[ \frac{1+dx}{x} \ln \frac{1+dx}{x} - \frac{1-dx}{x} \ln \frac{1-dx}{x} + 2d(\ln x - 1) \right] = 0. \quad (16)$$

故  $f(p)$  单调上升且以 0 为极限,  $f(p) < 0$ .  $\square$

由定理 1 可得,  $E_2[K(\varepsilon)] < E_1[K(\varepsilon)]$ , 说明剪枝策略实施后, 情况 1 中算法获得相同计算精度所需的迭代次数减少, 即局部开发能力增强.

**定理 2** 剪枝策略能改善算法局部开发能力的充分(非必要)条件是  $d < kp$ . 其中:  $k \approx 2.5$ , 且满足

$$k \ln(k^2 - 1) + \ln \frac{k+1}{k-1} = 2k. \quad (17)$$

**证明** 由定理 1 可知, 情况 1 中剪枝策略能改善算法局部开发能力这一结论是无条件的, 因而只需讨论情况 2 即可.

对于情况 2,  $d > p$ ,  $\overline{P_3P_4}$  内粒子  $K(\varepsilon)$  的期望值

表示为

$$E_2'[K(\varepsilon)] = \frac{1}{2d} \int_{p-d}^{p+d} (a \ln|x| + b) dx = \frac{1}{2d} \int_0^{d+p} (a \ln x + b) dx + \frac{1}{2d} \int_0^{d-p} (a \ln x + b) dx. \quad (18)$$

若要改善局部开发能力, 则须满足  $E_2'[K(\varepsilon)] < E_1[K(\varepsilon)]$ , 经整理并化简即可得到式 (17). 该式无闭式解, 数值解为  $k \approx 2.5043$ .  $\square$

综上所述, 剪枝策略在改善全局探索能力的同时, 通过选择合适的标准差  $\delta$  亦可提高局部开发能力.

### 2.3 标准 NPSO 算法及流程

本文将采用标准高斯分布形式的插枝方式作为标准 NPSO, 即  $\delta = 1$ . 标准 NPSO 的伪代码如下:

Set iterations  $M$ , population  $N$ , dimensions  $D$  and  $\alpha$ ; Initialize swarm  $X$ , find  $P$  and  $G$ .

```

for iter = 0, iter < M, iter ++ do
     $\delta = 1$ .
    for  $i = 0, i < N, i ++ do$ 
         $\xi_a = N(0, 1), \xi_b = N(0, 1)$ .
        for  $j = 0, j < D, j ++ do$ 
             $\xi = N(0, 1)$ ,
             $\mu = (\xi_a P[i] + \xi_b G) / (\xi_a + \xi_b)$ ,
             $X[i][j] = \mu + \alpha \xi (\mu - X[i][j])$ .
        end for
        if  $f(X[i]) < f(G)$ , then
             $G = X[i]$ .
            for  $k = 0, k < D, k ++ do$ 
                 $X[i][k] = P[i][k] = N(G[k], \delta)$ .
            end for
        else if  $f(X[i]) < f(P[i])$ , then
             $P[i] = X[i]$ .
        end if
    end for
end for

```

其中:  $X$  和  $P$  为二维数组, 保存粒子群位置和粒子历史最优位置;  $G$  为一维数组, 保存群体最优位置.

应注意上述流程针对的是无约束问题的优化. 对于带约束的问题, 可采取罚函数法处理相关约束, 即将约束项作为附加项归并到目标函数的计算, 除此之外算法流程与上述流程保持一致.

### 3 插枝方差 $\delta$ 的实验分析

为了进一步分析剪枝策略的特性并验证理论分

析的结论, 除了标准 NPSO, 本节提出一种插枝方差逐渐减小的策略, 记为 NPSO\*. NPSO\* 将  $\delta^2$  设置为粒子原始位置与粒子群中心  $M$  点各维度的偏差

$$\text{NPSO}^*: X_{id}^* = N(P_{id}, |M_d - P_{id}|), \quad (19)$$

其中  $M_d = \frac{1}{N} \sum_{i=1}^N X_{id}$ . 相应的标准 NPSO 的插枝方

$$\text{NPSO}: X_{id}' = N(P_{id}, 1). \quad (20)$$

在迭代过程中, NPSO\* 的插枝方差随着群体位置的变化而自适应调整. 一般情况下, 若粒子群趋向于收敛, 则迭代后期将接近 0, 但标准 NPSO 在整个过程中固定方差为 1.

通过实验考察两种策略的特性. 以 Sphere( $f_1$ )、Rosenbrock( $f_2$ )、Rastrigin( $f_3$ ) 和 Weierstrass( $f_4$ ) 这 4 个标准测试函数为优化对象, 最优解均为 0. 统一设置参数  $\alpha$  为 0.4, 函数及初始化范围设置为标准形式. 实验以经典 PSO 和 QPSO 两种算法为性能比较对象, 参数参照 Poli<sup>[15]</sup> 和 Sun<sup>[2]</sup> 的论文设置: 经典 PSO 参数  $c_1$  和  $c_2$  为 2.0, 惯性权重系数从 0.9 线性下降到 0.4, QPSO 的扩张-收缩系数从 1.0 线性下降到 0.5. 在所有算法中, 粒子数量为 40, 迭代次数为 4000, 优化函数的维度均为 30.

程序采用 C++ 编制, 在 VS 2010 环境下进行编译, CPU 为 Xeon E3-1230 3.2GHz. 实验重复 30 次, 其中最优化、平均解、最差解和各算法总消耗的时间记录于表 1.

在表 1 中: 每个函数的最好优化结果均用粗体标出,  $f_1$  和  $f_2$  为单峰函数, 而  $f_3$  和  $f_4$  为多峰函数. 由表 1 可知: 相比于 PSO 和 QPSO, NPSO\* 对于单峰函数的优化性能有大幅改善, 但对于多峰函数的改善一般, 甚至有所退步; 而标准 NPSO 对 4 个函数都有较好的改善, 尤其是对于多峰函数  $f_3$  和  $f_4$ , 但是对单峰函数的改善不如 NPSO\*. 由第 2.2 节的理论分析可知, 这一结果是可以预见的. 在进化中, NPSO\* 的插枝方差随着迭代的进行不断减小, 可视为满足定理 2 的条件, 从而能显著提升单峰函数性能. 但是对于多峰函数, 插枝方差的减小使粒子全局探索能力下降, 跳出陷入局部的能力下降. 标准 NPSO 保证了粒子全局探索能力, 但牺牲了部分局部开发能力, 在迭代后期有更大概率不能满足定理 2 的条件.

表 1 算法对  $f_1 \sim f_4$  的优化结果和时间消耗

Algorithm	$f_1$			$f_2$			$f_3$			$f_4$			t/s
	min	Av	max	min	Av	max	min	Av	max	min	Av	max	
PSO	4.91 e-28	8.36 e-24	2.21 e-22	6.567	81.67	329.1	16.91	27.19	44.77	<b>0</b>	0.03385	0.9840	<b>249</b>
QPSO	2.89 e-27	7.52 e-27	1.87 e-26	19.61	114.2	562.9	13.92	19.43	34.82	36.00	38.52	40.75	304
NPSO*	<b>1.28 e-93</b>	<b>1.71 e-85</b>	<b>3.93 e-84</b>	<b>4.28 e-05</b>	<b>11.90</b>	<b>78.67</b>	71.63	168.3	237.7	0.3354	5.521	39.83	298
NPSO	1.23 e-29	5.83 e-28	4.03 e-27	0.2699	48.36	177.6	<b>0.9949</b>	<b>3.520</b>	<b>12.93</b>	1.41 e-11	<b>3.73 e-06</b>	<b>4.44 e-05</b>	277

就算法计算消耗而言, 由于高斯分布数值的生成比均匀分布更复杂, NPSO 花费时间比 PSO 多了大概 10%, 但比 QPSO 略少, 4 种算法差别不大。

此外, 记录实验中不同算法单个粒子的运动轨迹如图 3 所示。

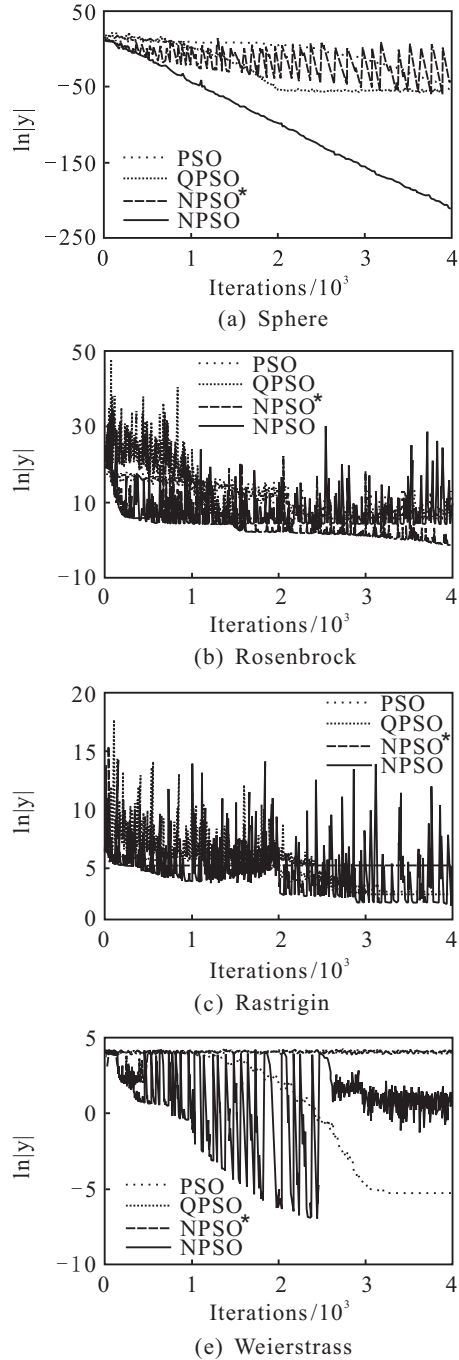


图 3 各算法优化过程中的粒子轨迹

在图 3 中, 纵坐标为目标函数绝对值的自然对数。

由图 3 能更清晰地看出 NPSO 的特点: NPSO\* 和标准 NPSO 粒子的跳动幅度远大于 PSO 和 QPSO, 除了随机数造成的粒子跳动, 剪枝策略将引入更大的随机扰动。由于插枝方差控制策略的不同, NPSO\* 的扰动随着迭代的进行将减弱, 而标准 NPSO 将保持稳定的剧烈扰动。

由图 3 可知: 在 Sphere 函数中, PSO 在 0~1 800 次迭代, QPSO 在 2 000~4 000 次迭代时的表现即为 burst 现象; 在 Rastrigin 函数中, PSO、QPSO 和 NPSO\* 都陷入了早熟; 在 Weierstrass 函数中, PSO 陷入了早熟, 而 QPSO 和 NPSO\* 都存在 burst 状态, 标准 NPSO 未出现早熟或 burst 现象。

若忽略计算机离散取值造成的计算误差, 由高斯分布的特性可知, 无论某局部最优点的峰有多深, 标准 NPSO 的粒子都能跳出该区域, 其粒子在局部最优早熟的概率为 0。

综上所述, 标准 NPSO 比经典 PSO 和 QPSO 的性能提升明显, 全局探索能力显著增强, 改善了粒子早熟和 burst 问题。但通过与 NPSO\* 的比较可知, 其局部开发能力仍有改进的空间。

#### 4 算法对比实验

为了更充分地检验标准 NPSO 算法的性能, 下面将考察其对 CEC 2005 Benchmark 函数<sup>[17]</sup>中的  $F_1 \sim F_{12}$  的优化表现, 比较算法分别为经典 GBBPSO、PSO、QPSO 和高斯量子粒子群 (GQPSO) 算法。前 5 个函数为单峰函数, 后 7 个为多峰函数。

PSO 和 QPSO 算法参数设置同上, GBBPSO 和 GQPSO 按照 Kennedy<sup>[1]</sup>和 Coelho<sup>[16]</sup>的文献进行设置。算法粒子数量为 40, 迭代次数为 4 000, 函数的维度均为 30。每个函数运行 50 次求解过程, 50 个运行结果的平均值见表 2, 最优值见表 3。表 2 和表 3 中每个函数的最好优化结果均用粗体标出。

由表 2 和表 3 可知, 在平均值上, 标准 NPSO 获得了 9 个函数的最优解, 而在最优值上获得了 10 个最优解。NPSO 对  $F_2$ 、 $F_4$  和  $F_8$  这 3 个函数的优化结果远远优于其他算法, 而在结果并非最优的几个函数中差距较小。总体而言, 各算法都保持较好的求解稳定性, NPSO 对算法总体性能的提升较为显著, 而 GBBPSO 和 GQPSO 两种改进与原算法相比也有一定的性能提升。

表 2 各算法对 CEC 2005 Benchmark 函数的优化结果平均值

Algorithm	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$
PSO	1.1e-23	1.5e+02	2.9e+07	1.4e+02	5.6e+03	4.0e+02	2.4e-02	1.8e-12	3.0e+01	1.1e+02	<b>4.2e-01</b>	7.6e+03
GPSO	3.4e-27	2.5e-01	6.0e+06	2.3e-01	4.8e+03	4.4e+02	2.3e-02	1.3e-01	5.6e+01	1.5e+02	1.8e+00	1.9e+04
QPSO	3.2e-25	1.9e+01	8.3e+06	1.3e+01	3.2e+03	4.5e+02	<b>2.5e-02</b>	1.9e+01	1.7e+01	7.3e+01	3.9e+01	1.0e+06
GQPSO	2.1e-25	7.4e+00	4.5e+06	6.9e+00	2.3e+03	2.7e+02	3.2e-02	1.7e-01	1.9e+01	<b>6.0e+01</b>	3.6e+01	8.7e+05
NPSO	<b>1.3e-27</b>	<b>6.8e-07</b>	<b>6.1e+05</b>	<b>5.3e-06</b>	<b>2.2e+03</b>	<b>8.9e+01</b>	3.2e-02	<b>4.2e-14</b>	<b>3.5e+00</b>	2.3e+02	3.5e+00	<b>2.3e+03</b>

表 3 各算法对 CEC 2005 Benchmark 函数的优化结果最优值

Algorithm	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$
PSO	3.3 e-26	4.0 e+01	1.2 e+07	2.2 e+01	3.5 e+03	4.0 e+00	2.8 e-05	8.5 e-13	1.8 e+01	6.1 e+01	5.5 e-05	<b>2.4 e+02</b>
GPSO	5.0 e-28	4.2 e-02	5.7 e+05	2.8 e-02	1.5 e+03	6.0 e+00	1.9 e-10	7.5 e-13	2.7 e+01	6.7 e+01	8.2 e-03	9.2 e+02
QPSO	1.2 e-25	2.3 e+00	1.8 e+06	3.2 e+00	6.5 e+02	2.2 e+01	7.4 e-03	1.0 e+01	8.9 e+00	<b>2.5 e+01</b>	3.7 e+01	8.2 e+05
GQPSO	1.1 e-25	1.0 e+00	1.2 e+06	1.3 e+00	1.1 e+03	2.1 e+01	9.8 e-03	7.5 e-14	1.3 e+01	2.7 e+01	3.2 e+01	7.4 e+05
NPSO	<b>3.1 e-28</b>	<b>6.7 e-08</b>	<b>2.5 e+05</b>	<b>1.4 e-06</b>	<b>8.7 e+02</b>	<b>2.2 e-01</b>	<b>3.2 e-08</b>	<b>1.4 e-14</b>	<b>1.4 e-06</b>	1.5 e+02	<b>3.33 e-07</b>	5.64 e+02

## 5 结 论

本文提出了一种骨干粒子群算法 NPSO, 主要工作内容包括: 1) 在进化方程中引入 3 个高斯分布, 通过即时搜索域分析指出, 所提出方程改善了粒子多样性. 2) 提出了剪枝策略. 剪枝策略本质是一种突变机制, 能增强算法全局探索能力. 定理 2 给出了能增强算法局部开发能力的充分条件——插枝方差需小于一定值. 3) 对于 CEC 2005 Benchmark 函数和标准测试函数的实验表明, NPSO 算法解决了传统 PSO 中常出现的 burst 状态和早熟问题, 显著提升了算法性能.

为了进一步讨论剪枝策略的特性, 本文提出了一种插枝方差时变策略 NPSO\*. NPSO\* 算法相比于标准 NPSO 算法更接近定理 2 的条件. 实验结果证明, NPSO\* 局部开发能力显著增强, 但全局开发探索能力不如标准 NPSO, 这说明剪枝策略还有很大的优化潜力. 今后的工作将包括: 1) 标准 NPSO 在迭代后期将不满足定理 2 的条件, 对单峰函数性能改善效果将减弱, 需调整优化插枝方差  $\delta$ , 以进一步增强局部开发能力; 2) 研究将剪枝策略引入经典 PSO, 由于经典 PSO 粒子带有速度项, 剪枝的同时需解决速度修正问题.

## 参考文献(References)

- [1] Kennedy J. Bare bones particle swarms[C]. Proc of the Swarm Intelligence Symposium. Indiana, 2003: 80-87.
- [2] Sun J, Fang W, Wu X, et al. Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection[J]. Evolutionary Computation, 2012, 20(3): 349-393.
- [3] Blackwell T. A study of collapse in bare bones particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2012, 16(3): 354-372.
- [4] Yao J, Han D. Improved barebones particle swarm optimization with neighborhood search and its application on ship design[J]. Mathematical Problems in Engineering, 2013: 1-13.
- [5] Zhang H, Kennedy D D, Rangaiah G P, et al. Novel bare-bones particle swarm optimization and its performance for modeling vapor-liquid equilibrium data[J]. Fluid Phase Equilibria, 2011, 301(1): 33-45.
- [6] Zhang Y, Gong D, Geng N, et al. Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects[J]. Applied Soft Computing, 2014(18): 248-260.
- [7] 史丽萍, 王攀攀, 胡泳军, 等. 基于骨干微粒群算法和支持向量机的电机转子断条故障诊断[J]. 电工技术学报,

2014, 29(1): 147-155.

- (Shi L P, Wang P P, Hu Y J, et al. Broken rotor bar fault diagnosis of induction motors based on bare-bone particle swarm optimization[J]. Trans of China Electricotechnical Society, 2014, 29(1): 147-155.)
- [8] Van Den Bergh F. An analysis of particle swarm optimizers[D]. Pretoria: Department of Computer Science, University of Pretoria, 2006.
- [9] 刘衍民, 隋常玲, 赵庆祯. 基于  $K$ -均值聚类的动态多种群粒子群算法及其应用[J]. 控制与决策, 2011, 26(7): 1019-1025.  
(Liu Y M, Sui C L, Zhao Q Z. Dynamic multi-swarm particle swarm optimizer based on  $K$ -means clustering and its application[J]. Control and Decision, 2011, 26(7):1019-1025.)
- [10] 范成礼, 邢清华, 范海雄, 等. 带收敛因子的变邻域粒子群算法[J]. 控制与决策, 2013, 29(4): 696-700.  
(Fan C L, Xing Q H, Fan H X, et al. Particle swarm optimization and variable neighborhood search algorithm with convergence criterions[J]. Control and Decision, 2013, 29(4): 696-700.)
- [11] Al-Rifaie M M, Blackwell T. Bare bones particle swarms with jumps[M]. Berlin Heidelberg: Springer, 2012: 49-60.
- [12] 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): 1253-1257.  
(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): 1253-1257.)
- [13] Zabinsky Z B, Smith R L. Pure adaptive search in global optimization[J]. Mathematical Programming, 1992, 53(1/2/3): 323-338.
- [14] Sun J, Wu X, Palade V, et al. Convergence analysis and improvements of quantum-behaved particle swarm optimization[J]. Information Sciences, 2012, 193: 81-103.
- [15] Poli R, Kennedy J, Blackwell T. Particle swarm optimization[J]. Swarm Intelligence, 2007, 1(1): 33-57.
- [16] Coelho L D S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems[J]. Expert Systems with Applications, 2010, 37(2):1676-1683.
- [17] Suganthan P, Hansen N, Liang J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[R]. Singapore: Nanyang Technological University, 2005.

(责任编辑: 闫 妍)