

文章编号: 1001-0920(2011)07-1083-04

## 基于混沌变异的自适应双粒子群优化

吴定海, 张培林, 李 胜, 徐 超, 王怀光

(机械工程学院, 石家庄 050003)

**摘要:** 针对粒子群优化在解决高维优化问题时收敛性差、搜索效率不高的问题, 在对粒子群优化算法收敛性分析的基础上, 提出了混沌变异对极值进行扰动的方法, 以增强算法摆脱局部最优解的能力. 采用自适应惯性权重和局部邻域搜索保持较高的局部搜索性能, 并结合双粒子群协同进化的方法, 综合平衡优化算法的全局搜索和局部搜索能力. 通过对 4 个典型测试函数进行的对比实验, 表明了所提出的算法能大大提高粒子群优化的搜索效率和收敛精度.

**关键词:** 粒子群; 混沌; 自适应; 极值扰动

中图分类号: TP18

文献标识码: A

## Adaptive double particle swarms optimization algorithm based on chaotic mutation

WU Ding-hai, ZHANG Pei-lin, LI Sheng, XU Chao, WANG Huai-guang

(Ordnance Engineering College, Shijiazhuang 050003, China. Correspondent: WU Ding-hai, E-mail: wd81@163.com)

**Abstract:** Considering the bad convergence and search efficiency of particle swarm optimization(PSO) when applied to multi-dimensional optimization problem, an algorithm of adaptive chaotic variation particle swarm optimization(ACPSO) is proposed after analyzing the convergence of PSO. The proposed algorithm uses the technique of double subpopulation cooperative evolution. One subpopulation uses a new disturbed extremum method of chaotic mutation to improve the global convergence ability, and the other uses the adaptive inertia weight and neighborhood information to enhance the local search ability. The simulation results of four typical functions show that the proposed algorithm can attain better optimization performance.

**Key words:** particle swarm optimization; chaotic; adaptive; disturbed extremum

### 1 引言

Kennedy 等人受鸟群和鱼群搜索食物时个体与群体进行信息交流和共享的协作方式的启发, 提出一种基于群体智能的粒子群优化算法(PSO)<sup>[1]</sup>. 该算法具有收敛速度快、算法简单、调节参数少等优点, 在多目标优化、模式分类等领域得到了广泛应用<sup>[2-3]</sup>.

微粒群算法在搜索过程中易陷入局部最优, 因此如何克服早熟收敛一直是粒子群优化算法研究的重点问题. Eberhart 等人<sup>[4]</sup>将惯性权重  $\omega$  引入速度更新方程, 体现了粒子先前速度对当前速度的影响程度, 在一定程度上起到了均衡全局搜索与局部搜索的能力. 文献 [5] 从粒子的速度出发, 通过引入激活因子来保持粒子活性以提高搜索能力. 文献 [6] 分析了不同的信息共享策略对搜索性能的影响, 指出粒子共享邻域个体信息的重要性.

基于以上研究, 本文从分析粒子群的收敛性出发, 为增强算法摆脱陷入局部最优解的能力, 保持较高的局部搜索性能, 提出一种基于混沌变异的自适应双粒子群优化算法(ACPSO), 并通过标准测试函数验证了该算法的有效性.

### 2 PSO 收敛性及影响分析

#### 2.1 PSO 收敛性分析<sup>[7]</sup>

微粒群的速度方程可改写为

$$v_i(t+1) = wv_i(t) + c_1r_1p_{best_i}(t) + c_2r_2g_{best_i}(t) - (c_1r_1 + c_2r_2)x_i(t). \quad (1)$$

当  $t \rightarrow \infty$  时,  $v(t) = 0$  且  $v(t+1) = 0$ . 由速度方程得

$$\lim_{t \rightarrow \infty} x^* = \frac{c_1r_1p_{best} + c_2r_2g_{best}}{c_1r_1 + c_2r_2}. \quad (2)$$

由于  $r_1, r_2$  服从均匀分布, 求期望观察得

收稿日期: 2010-04-23; 修回日期: 2010-06-24.

**作者简介:** 吴定海(1981—), 男, 博士生, 从事机械状态检测、信号处理等研究; 张培林(1955—), 男, 教授, 博士生导师, 从事机械状态检测、智能故障诊断等研究.

$$\lim_{t \rightarrow \infty} E(x^*) = E\left(\frac{c_1 r_1 p_{\text{best}} + c_2 r_2 g_{\text{best}}}{c_1 r_1 + c_2 r_2}\right) = \frac{c_1 p_{\text{best}} + c_2 g_{\text{best}}}{c_1 + c_2} = (1 - \alpha)p_{\text{best}} + \alpha g_{\text{best}}. \quad (3)$$

由上式可知,在标准微粒群中,粒子进化收敛的过程就是不断地向 $x^*$ 聚集的过程,且收敛的极值由自身极值 $p_{\text{best}}$ 和群体极值 $g_{\text{best}}$ 决定.如果所有粒子在向 $x^*$ 靠近过程中没有找到更优于 $g_{\text{best}}$ 的位置,则此时粒子的进化将处于停滞状态,整个种群陷入局部最优解 $x^*$ .

从微粒群的收敛性分析可以看出,PSO在优化前期收敛速度很快,但在优化后期收敛速度变得缓慢,其主要原因是粒子群难以摆脱局部极值的影响.

## 2.2 惯性权重的影响分析

惯性权重对粒子的搜索能力具有较大影响.取值范围一般为 $[0,1]$ ,大的惯性权重能够较好地进行全局搜索,小的惯性权重则局部搜索性能较好.对于每个粒子而言,当粒子与全局最优解相距较远时,需提高飞行速度进行全局搜索,而当距离全局最优解较近时则降低飞行速度,进行局部细化搜索.实际上,惯性权重的设置是对全局搜索与局部搜索性能的平衡.

采用固定权重和线性递减惯性权重的策略,对于每一个粒子,不管它的距离远近,都采用同样的惯性权重不可避免地会在全局最优解附近发生“交叉振荡”现象(如图1所示),从而影响收敛速度.

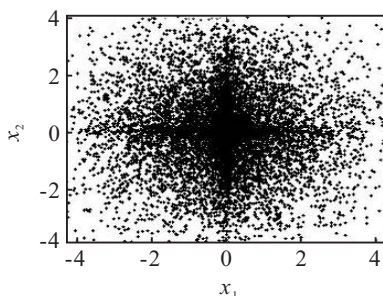


图1 交叉振荡现象

## 3 自适应混沌双粒子群优化

### 3.1 双种群协同进化

针对以上对微粒群的分析,本文提出一种基于混沌变异的自适应双种群粒子群优化算法.采用2个独立的粒子群进行协同优化,其中种群1采用标准的微粒群算法(惯性权重取0.793),充分利用标准粒子群在搜索前期效率高的特点执行全局搜索;种群2采用自适应惯性权重和局部邻域搜索的方法执行局部细化搜索,以提高搜索精度.将2个种群在搜索过程中所发现的最优粒子作为全局最优解.

### 3.2 混沌变异全局搜索种群

从以上对微粒群收敛性的分析可知,各个粒子在进化过程中均向 $p_{\text{best}}$ 和 $g_{\text{best}}$ 学习,迭代初期种群的多样性较好,随着迭代次数的增加,粒子逐渐从 $p_{\text{best}}$ 向 $g_{\text{best}}$ 聚集,种群的多样性迅速下降,尤其在搜索后期,相邻迭代次数之间的 $p_{\text{best}}$ 或 $g_{\text{best}}$ 的值已经非常接近,粒子将逐渐停滞,陷入局部最优.此时,调整个体极值 $p_{\text{best}}$ 和全局极值 $g_{\text{best}}$ ,引导所有粒子飞向新的位置,从 $x^*$ 迁移聚集到一个新位置,经历新的搜索路径和领域,增加发现更优解的概率,摆脱局部极值.

为了使PSO具有更高的搜索效率,有效摆脱局部最优,本文提出一种新的基于混沌变异因子的扰动策略.

如图2所示,混沌变量具有以下几个显著特点:1)对初值具有敏感性;2)伪随机性;3)遍历性;4)规律性.因此,引入混沌变量可以避免重复搜索,提高搜索效率.

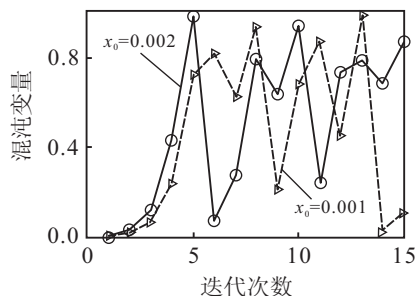


图2 Logistic混沌系统中相邻两点运动轨迹分析

本文引入混沌变异因子,利用全局迭代最优值的停滞步数作为变异的触发条件,采用混沌变异因子对个体历史和全局最优进行混沌变异,对粒子的搜索方向加以引导和改进.混沌变异实现方法如下:

```

if  $k > 5$ ; then
     $x = \mu \cdot x \cdot (1 - x)$ ,  $y = \mu \cdot y \cdot (1 - y)$ ,
     $p_{\text{best}} = p_{\text{best}} \cdot x$ ,  $g_{\text{best}} = g_{\text{best}} \cdot y$ 
end

```

首先,随机产生向量 $x$ 和 $y$ ,当优化停滞步数大于5时,由Logistic映射进行迭代,产生混沌变量;然后,进行混沌变异,更新粒子的历史最优解和全局最优解的值,以提高全局搜索能力.

### 3.3 自适应邻域局部搜索种群

在粒子群的搜索过程中,经常变异必然影响粒子群的搜索效率,过度变异将导致混乱.因此,引入另一个种群进行协同进化,以加强局部搜索能力,从而平衡优化算法的全局和局部搜索能力.

根据以上关于惯性权重对搜索性能影响的分析,

本文对该种群采用自适应的惯性权重, 对每一个粒子进行有差别对待, 以粒子的适应度值作为评价指标. 当粒子的适应度值与群体最优值差别较大时, 采用较大的惯性权重, 以加快搜索速度; 当粒子的适应度值与群体最优值差别较小时, 采用较小的惯性权重, 在最优值周围进行细化搜索, 以提高搜索的精度.

动态自适应惯性权重设置如下:

$$\omega_i^k = e_i^{-\lambda}, \tag{4}$$

$$\lambda = \alpha_i^k / \alpha_i^{k-1}, \tag{5}$$

$$\alpha_i^k = |f(x_i^k) - f(g_{\text{best}}^k)|. \tag{6}$$

式中:  $f(x_i^k)$  为第  $i$  个粒子在第  $k$  次迭代时所对应的适应度函数值,  $f(g_{\text{best}}^k)$  为第  $k$  次迭代时群体最优适应度函数值.

采用动态自适应惯性权重在一定程度上可避免在全局最优解附近发生“交叉振荡”现象.

为加强局部搜索能力, 对引导粒子学习的拓扑邻域进行改进, 速度的更新公式更改为

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 \text{rand}_1 [p_{\text{best}_{i,j}}(t) - x_{i,j}(t)] + c_2 \text{rand}_2 \left[ \sum_{k=1}^m l_{\text{best}_k}(t) - x_{i,j}(t) \right]. \tag{7}$$

式中  $m$  是一个动态浮动的值, 在粒子群优化搜索过程

中寻找邻域适应度值优于自身的粒子作为学习对象.

所谓邻域搜索的方法, 就是粒子除向自身历史最优学习外, 还向邻域的最优粒子学习. 在初始阶段, 将每个粒子的邻域定义为粒子本身, 随着迭代次数的增加, 将邻域的搜索范围扩大到包含所有粒子, 邻域极值为种群的最优值. 邻域搜索方法能够增强粒子在邻域的细化搜索能力.

## 4 仿真实验及结果分析

### 4.1 测试函数名称及参数设置

为了综合测试 ACP SO 算法的优化性能, 本文采用了 4 种标准的 Benchmark 测试函数, 即 Rosenbrock, Rastrigrin, Griewank 和 Ackly (表达式如表 1 所示), 比较了标准粒子群 (PSO), 基于线性递减惯性权重的粒子群 (LPSO) 和本文提出的 ACP SO 的优化性能.

参数设置如下: ACP SO 两种群规模取为 20, 维数取 100, 每个函数独立运行 50 次. PSO 和 LPSO 种群规模为 40, 惯性权重因子取 0.7, 学习因子均取 2. LPSO 的惯性权重取初值 0.9, 末值 0.4, 迭代次数超过 1 500 代后, 惯性权重置为初始值. 仿真实验时, 最大迭代次数为 3 000, 为比较算法优化效率和精度, 评价参数为平均最优值 (Mean Best), 标准方差 (St.Dev), 最小值 (Min), 收敛率 (Ras). 收敛率为最大步数内能够达到收敛精度的概率.

表 1 标准测试函数

| 函数名称       | 函数表达式  | 维数  | 定义域             | 收敛精度 |
|------------|--|-----|-----------------|------|
| Rosenbrock | $f_3 = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$  | 100 | $[-100, 100]$   | 100  |
| Rastrigrin | $f_2 = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$  | 100 | $[-5.12, 5.12]$ | 100  |
| Griewank   | $f_4 = \frac{1}{400} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$   | 100 | $[-600, 600]$   | 0.1  |
| Ackly      | $f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$ | 100 | $[-30, 30]$     | 0.05 |

### 4.2 实验结果分析

Rosenbrock 是一个典型的单模态优化测试函数, PSO 采用固定的惯性权重, 前期具有较高的搜索效率 (如图 3(a) 所示), 但在搜索的中后期, 难以进行局部细化搜索, 收敛速度减慢. LPSO 前期搜索速度慢, 中后期局部搜索能力较强, 但是, 一旦陷入局部极小, 则很难再跳出. 所以对于高维单模态优化问题并不能取得很好的效果. ACP SO 采用双粒子群进行协同优化搜索, 对于单模态问题, 搜索前期种群 1 采用固定的惯性权重, 具有较快的搜索效率, 在搜索的后期种群 2 的邻域搜索使局部搜索能力得到发挥, 通过信息的共享进行协同进化, 取得了很好的效果.

对于 Rastrigrin, Griewank 和 Ackly 标准多模态测试函数, Rastrigrin 主要用于考察优化算法的全局搜索性能. Ackly 是一个有很多深谷的多模态函数, 其局部极小点均匀地分布在整个搜索空间内, 该函数用于测试优化算法跳出局部最优的能力. 如图 3(b) 和图 3(c) 所示, 相比于 PSO 和 LPSO 算法, ACP SO 凭借混沌变异因子, 使每次变异都能指向不同的方向, 增大了粒子群跳出局部极小点的可能性, 取得了较好的收敛效果. Griewank 函数是一个各维显著相互影响的多模态函数, 具有无穷多个极值点, 用于综合考察优化算法对全局与局部搜索能力上的平衡. 如图 3(d) 所示, ACP SO 采用混沌变异因子改变了引导粒子进化的方向信息, 同时结合种群 2 的自适应惯性权重和局部邻

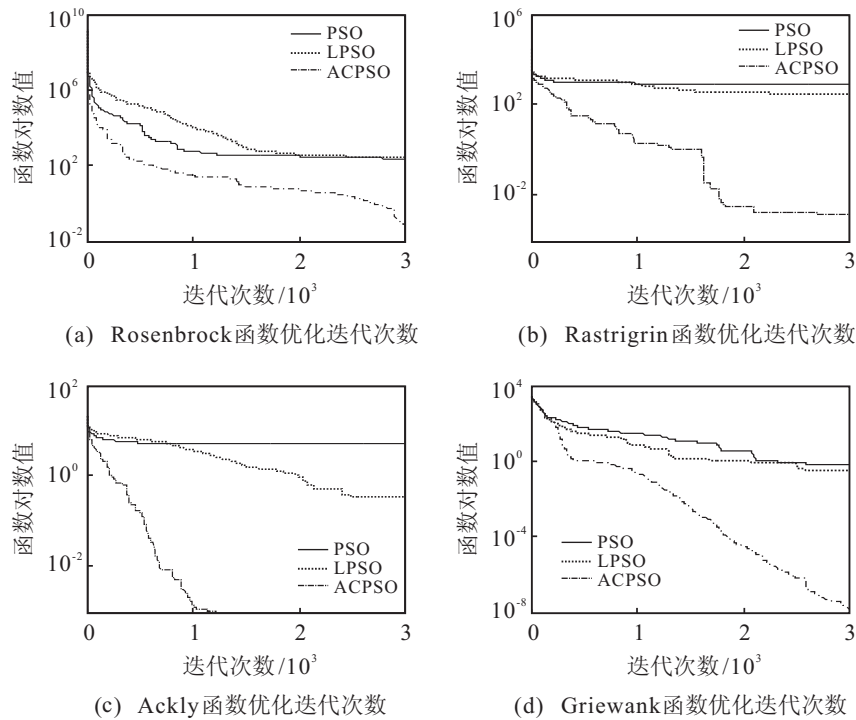


图3 标准测试函数优化效果比较

表2 ACPSO算法与其他算法在Benchmark优化问题上的比较

| Function   | Algorithm | Mean Best   | St.Dev     | Min       | Ras   |
|------------|-----------|-------------|------------|-----------|-------|
| Rosenbrock | PSO       | 239.440 4   | 44.770 5   | 180.164 1 | 0/50  |
|            | LPSO      | 155.879 2   | 87.441 2   | 38.532 5  | 8/50  |
|            | ACPSO     | 56.432 1    | 28.358 3   | 9.462 8   | 46/50 |
| Rastrigrin | PSO       | 202.974 8   | 16.086 5   | 165.179 7 | 0/50  |
|            | LPSO      | 169.143 1   | 11.687 7   | 148.008 3 | 0/50  |
|            | ACPSO     | 0.005 592   | 0.001 922  | 2.367e-8  | 50/50 |
| Ackly      | PSO       | 5.020 5     | 0.793 1    | 4.263 7   | 0/50  |
|            | LPSO      | 0.900 3     | 0.001 62   | 4.1e-3    | 37/50 |
|            | CPSO      | 1.267 3e-15 | 5.493 0e-9 | 0         | 50/50 |
| Griewank   | PSO       | 23.057 4    | 8.342 7    | 0.074 1   | 9/50  |
|            | LPSO      | 11.146 9    | 5.363 3    | 0.017 76  | 17/50 |
|            | ACPSO     | 0.000 652   | 0.050 6    | 0         | 49/50 |

域搜索进行局部细化搜索, 具有较好的全局搜索和局部搜索能力。

表2列出的实验数据表明, 对于高维优化问题, 不论是单模态或多模态优化, PSO和LPSO的寻优能力和收敛速度都明显减弱, 并不能很好地满足工程应用的需要. 本文提出的自适应混沌粒子群优化算法采用引入混沌变异因子的方法增强了粒子群跳出局部最优的能力, 利用双种群协同进化提高了局部细化搜索能力, 并通过实现信息的共享实现协同进化, 具有很好的全局收敛能力, 其搜索成功率和收敛精度均明显高于PSO和LPSO.

## 5 结 论

本文通过对微粒群算法收敛性的分析, 指出了微粒群易陷入局部最优的根本原因, 并在此基础上

提出了一种基于混沌变异的自适应双粒子群优化方法. 该方法采用2个种群进行协同进化和分工合作, 种群1通过运用混沌变量对极值进行扰动, 增强了全局搜索能力; 种群2通过运用自适应惯性权重和局部邻域搜索, 提高了局部细化搜索性能. 标准测试函数的实验和比较结果验证了本文提出的ACPSO算法能够很好地平衡全局搜索和局部搜索能力, 可以大大改善微粒群优化算法的性能, 在解决高维优化问题上具有很好的收敛速度和搜索精度.

## 参考文献(References)

- [1] Clerc M, Kennedy J. The particle swarm: explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.