

# 自适应变异的粒子群优化算法

吕振肃, 侯志荣

(兰州大学信息科学与工程学院, 甘肃兰州 730000)

**摘 要:** 本文提出了一种新的基于群体适应度方差自适应变异的粒子群优化算法(AMPSO). 该算法在运行过程中根据群体适应度方差以及当前最优解的大小来确定当前最佳粒子的变异概率, 变异操作增强了粒子群优化算法跳出局部最优解的能力. 对几种典型函数的测试结果表明: 新算法的全局收搜索能力有了显著提高, 并且能够有效避免早熟收敛问题.

**关键词:** 粒子群; 自适应变异; 优化; 早熟收敛

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112 (2004) 03-0416-05

## Particle Swarm Optimization with Adaptive Mutation

LÜ Zhen-su, HOU Zhi-rong

(School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu 730000, China)

**Abstract:** A new adaptive mutation particle swarm optimizer (AMPSO), which is based on the variance of the population's fitness is presented. During the running time, the mutation probability for the current best particle is determined by two factors: the variance of the population's fitness and the current optimal solution. The ability of particle swarm optimization algorithm (PSO) to break away from the local optimum is greatly improved by the mutation. The experimental results show that the new algorithm not only has great advantage of convergence property over genetic algorithm and PSO, but also can avoid the premature convergence problem effectively.

**Key words:** particle swarm; adaptive mutation; optimization; premature convergence

## 1 引言

粒子群优化算法 PSO (Particle Swarm Optimization) 是由 Eberhart 博士和 Kennedy 博士发明的一种新的全局优化进化算法, 它源于对鸟类捕食行为的模拟<sup>[1,2]</sup>. 作为一种重要的优化工具, 粒子群优化算法已经成功地用于系统辨识<sup>[3]</sup>、神经网络训练<sup>[4]</sup>等领域.

与其它全局优化算法(如遗传算法)一样, 粒子群优化算法同样存在早熟收敛现象, 尤其是在比较复杂的多峰搜索问题中. 目前解决这一问题的主要方法是增加粒子群的规模, 虽然对算法性能有一定改善, 但同样存在缺陷: 一是不能从根本上克服早熟收敛问题; 二是会大量增加算法的运算量. 本文将提出一种新的基于群体适应度方差自适应变异的粒子群优化算法(AMPSO). 该算法根据群体适应度方差以及当前最优解的大小来确定当前最佳粒子的变异概率. 实验结果表明: 与遗传算法和粒子群优化算法相比, 本文算法的全局收敛性得到了显著提高, 能有效避免粒子群优化算法中的早熟收敛问题.

## 2 粒子群优化算法及其早熟收敛问题

与遗传算法类似, 粒子群优化算法同样基于群体(这里称作粒子群)与适应度. 粒子群的个体(这里称作粒子)代表问题的一个可能解. 每个粒子具有位置和速度两个特征. 粒子位置坐标对应的目标函数值即可作为该粒子的适应度. 算法通过适应度来衡量粒子的优劣. 算法首先初始化一群随机粒子, 然后通过迭代找到最优解. 在每一次迭代中, 粒子通过跟踪两个“极值”来更新自己: 一个是粒子本身所找到的最优解, 即个体极值  $pBest$ ; 另一个是整个粒子群目前找到的最优解, 称之为全局极值  $gBest$ .

粒子在找到上述两个极值后, 就根据下面两个公式来更新自己的速度与位置<sup>[1,2]</sup>:

$$V = w * V + c_1 * rand * (pBest - Present) + c_2 * rand * (gBest - Present) \quad (1)$$

$$Present = Present + V \quad (2)$$

其中,  $V$  是粒子的速度,  $Present$  是粒子的当前位置.  $rand$  是  $[0, 1]$  之间的随机数,  $c_1$  和  $c_2$  被称作学习因子. 通常,  $c_1 = c_2 = 2$ .

$w$  是加权系数,一般在 0.1 到 0.9 之间取值.文献[2]通过大量实验证明,如果  $w$  随算法迭代的进行而线性减小,将显著改善算法的收敛性能.设  $w_{\max}$  为最大加权系数,  $w_{\min}$  为最小加权系数,  $run$  为当前迭代次数,  $runMax$  为算法迭代总次数,则有:

$$w = w_{\max} - run * \frac{(w_{\max} - w_{\min})}{runMax} \quad (3)$$

更新过程中,粒子每一维的最大速率限制在  $v_{\max}$ ,粒子每一维的坐标也被限制在允许范围之内.同时,  $pBest$  与  $gBest$  在迭代过程中不断更新,最后输出的  $gBest$  就是算法得到的最优解.

粒子群优化算法一般采用实数编码,由于没有选择、交叉与变异等操作,算法结构相对简单,运行速度很快.但是,算法运行过程中,如果某粒子发现一个当前最优位置,其他粒子将迅速向其靠拢.如果该最优位置为一局部最优点,粒子群就无法在解空间内重新搜索,因此,算法陷入局部最优,出现了所谓的早熟收敛现象.

试验证明,粒子群优化算法无论是早熟收敛还是全局收敛,粒子群中的粒子都会出现“聚集”现象.要么所有粒子聚集在某一特定位置,要么聚集在某几个特定位置,这主要取决于问题本身的特性以及适应度函数的选择.下面将从理论上证明这个结论.粒子位置的一致等价于各粒子的适应度相同.因此,研究粒子群中所有粒子适应度的整体变化就可以跟踪粒子群的状态.为了定量描述粒子群的状态,下面先给出群体适应度方差的定义,同时也给出了粒子收敛的定义.

**定义 1** 设粒子群的粒子数目为  $n$ ,  $f_i$  为第  $i$  个粒子的适应度,  $f_{avg}$  为粒子群目前的平均适应度,  $\sigma^2$  为粒子群的群体适应度方差,则  $\sigma^2$  可以定义为:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n \left( \frac{f_i - f_{avg}}{f} \right)^2 \quad (4)$$

其中  $f$  是归一化定标因子,其作用是限制  $\sigma^2$  的大小.  $f$  可以取任意值,只需注意两个条件: 归一化后,整个粒子群  $|f_i - f_{avg}|$  的最大值不大于 1;  $f$  随算法的进化而变化.在本文算法中,  $f$  的取值采用如下公式:

$$f = \begin{cases} \max\{|f_i - f_{avg}|\}, \max\{|f_i - f_{avg}|\} > 1 \\ 1, \text{others} \end{cases} \quad (5)$$

**定义 1** 表明:群体适应度方差  $\sigma^2$  反映的是粒子群中所有粒子的“收敛”程度.  $\sigma^2$  越小,则粒子群趋于收敛;反之,粒子群则处于随机搜索阶段.

**定义 2** 设粒子群中某个粒子在  $t$  时刻的位置为  $x(t)$ ,  $p$  为搜索空间内的任意位置,则粒子收敛定义如下<sup>[5]</sup>:

$$\lim_{t \rightarrow +\infty} x(t) = p \quad (6)$$

该定义表明,粒子的收敛是指粒子最终停留在搜索空间内某一固定位置  $p$ .

**定理 1** 如果粒子群优化算法陷入早熟收敛或者达到全局收敛,粒子群中的粒子将聚集在搜索空间的一个或几个特定位置,群体适应度方差  $\sigma^2$  等于零.

**证明** 根据定义 2,粒子如果收敛,将停留在某一固定位置  $p$ .下面讨论如何确定收敛位置  $p$ .文献[5]通过严格的数

学推导,得出如下结论:

$$\lim_{t \rightarrow +\infty} x(t) = (1 - a)y + ay \quad (7)$$

其中  $a = c_1 / (c_1 + c_2)$ ,  $c_1$  与  $c_2$  是式(1)中的学习因子.  $y$  表示粒子当前的个体极值,  $y$  表示粒子群当前的全局极值.若  $c_1 = c_2 = 2$ ,则式(7)变为:

$$\lim_{t \rightarrow +\infty} x(t) = \frac{y + y}{2} \quad (8)$$

文献[5]在推导式(7)时假设  $y$  和  $y$  都固定不变.从式(8)可以看出,如果  $y$  和  $y$  都固定不变,则粒子的收敛位置  $p$  就是粒子个体极值与粒子群全局极值之间的中点位置.实际上,  $y$  和  $y$  一般都随时间变化而改变.此时,将式(8)改写为:

$$\lim_{t \rightarrow +\infty} x(t) = \lim_{t \rightarrow +\infty} \frac{y(t) + y(t)}{2} \quad (9)$$

根据粒子群优化算法的原理,粒子在位置更新与速度更新的过程中,如果发现新位置优于个体极值,则将个体极值设置为新位置;同样,如果新位置优于全局极值,则将全局极值设置为新位置.不失一般性,设粒子群最终找到的全局极值为  $p^*$ ,即:

$$\lim_{t \rightarrow +\infty} y(t) = p^* \quad (10)$$

显然,如果全局极值为  $p^*$ ,粒子将在式(2)的作用下向  $p^*$  靠拢,其间将不断更新个体极值  $y$ ,如果粒子没有发现比  $p^*$  更好的位置,那么其个体极值最终将等于  $p^*$ ,即:

$$\lim_{t \rightarrow +\infty} y(t) = p^* \quad (11)$$

将式(10)与式(11)代入式(9),可得:

$$\lim_{t \rightarrow +\infty} x(t) = p^* \quad (12)$$

式(12)表明,对于粒子群中的任意粒子,其最终收敛位置将是整个粒子群找到的全局极值.如果粒子群找到的全局极值只有一个,那么所有粒子都会“聚集”到该位置;如果全局极值不止一个,那么粒子将随机聚集在这几个全局极值位置.全局极值是所有粒子在算法运行过程中找到的最佳粒子位置,该位置并不一定就是搜索空间中的全局最优点.若该位置为全局最优点,则算法达到全局收敛;否则算法陷入早熟收敛.

根据粒子位置可计算相应的适应度,因此,当算法陷入早熟收敛或全局收敛时,粒子群中的粒子位于全局极值  $p^*$ .设  $f(x)$  为适应度函数,粒子适应度可由下式计算:

$$f_i = f(p^*), \quad i = 1, 2, \dots, n \quad (13)$$

根据定义 1,粒子群目前的平均适应度可由式(14)给出:

$$f_{avg} = \frac{1}{n} \sum_{i=1}^n f_i \quad (14)$$

将式(13)代入式(14),可得

$$f_{avg} = f(p^*) \quad (15)$$

将式(13)与式(15)代入式(4),可得  $\sigma^2 = 0$ .定理 1 得证.

定理 1 给出了粒子群优化算法收敛状态与群体适应度方差之间的关系.显然,仅凭群体适应度方差等于零不能区别早熟收敛与全局收敛,还须进一步判断算法此时得到的最优解是否为理论全局最优解或者期望最优解  $f_d$ .如果此时已经得到全局最优,则可认为算法达到全局收敛;反之,则表明算法陷入局部最优.

图 1 是粒子群优化算法求解某函数最小化问题全局收敛的适应度方差进化曲线,图 2 则是早熟收敛的适应度方差进

化曲线.

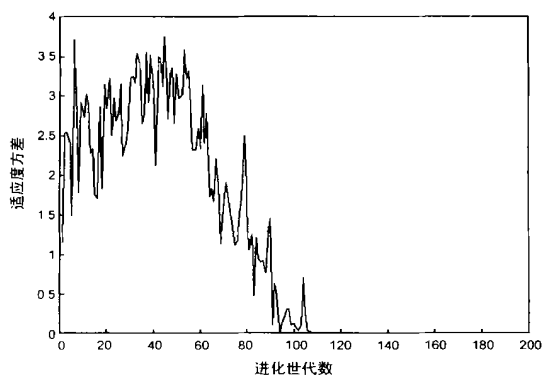


图 1 PSO 全局收敛的适应度方差进化曲线

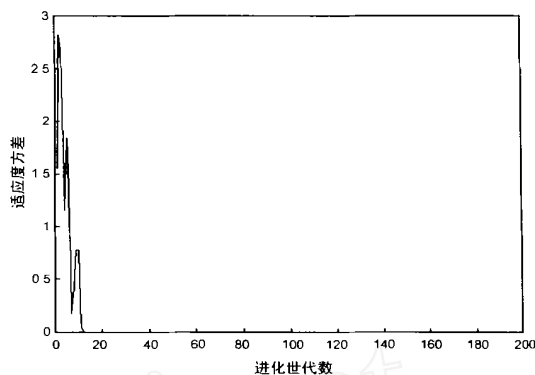


图 2 PSO 早熟收敛的适应度方差进化曲线

### 3 自适应变异的粒子群优化算法

从前面的分析可知,在粒子群优化算法运行过程中,如果群体适应度方差等于零,且此时得到的最优解不是理论最优解或者期望最优解  $f_d$ ,则粒子群陷入局部最优,算法将出现早熟收敛.因此,如果要克服早熟收敛问题,就必须提供一种机制,让算法在发生早熟收敛时,能够跳出局部最优,进入解空间的其它区域继续进行搜索,直到最后找到全局最优解.

根据式(1)和式(2),粒子下一时刻的位置由当前位置与当前速度共同决定,速度大小决定移动距离,速度方向决定粒子前进方向.根据式(1),粒子当前速度由三个因素决定:原来的速度、个体极值  $pBest$  与全局极值  $gBest$ .全局极值  $gBest$  是算法目前找到的最优解.如果算法出现早熟收敛,全局极值  $gBest$  一定是局部最优解.结合式(1),如果此时改变全局极值  $gBest$ (变异操作),就可以改变粒子的前进方向,从而让粒子进入其它区域进行搜索,在其后的搜索过程中,算法就可能发现新的个体极值  $pBest$  以及全局极值  $gBest$ .如此循环,算法就可以找到全局最优解.这就是本文将要提出的自适应变异机制的基本思想.

考虑到粒子在当前  $gBest$  的作用下可能发现更好的位置,因此新算法将变异操作设计成一个随机算子,即对满足变异条件的  $gBest$  按一定的概率  $p_m$  变异.  $p_m$  的计算公式如下:

$$p_m = \begin{cases} k, & \sigma^2 < \sigma_d^2 \text{ and } f(gBest) > f_d \\ 0, & \text{others} \end{cases} \quad (16)$$

其中,  $k$  可以取  $[0, 0.1, 0.3]$  之间的任意数值.  $\sigma_d^2$  的取值与实际问题的有关,一般远小于  $\sigma^2$  的最大值.  $f_d$  可以设置为理论最优值.这里考虑的是“最小化”情况.

对于  $gBest$  的变异操作,本文算法将采用增加随机扰动的方法,设  $gBest_k$  为  $gBest$  的第  $k$  维取值,是服从 Gauss  $(0, 1)$  分布的随机变量,则

$$gBest_k = gBest_k * (1 + 0.5 * \text{rand}) \quad (17)$$

综上所述,我们对粒子群优化算法进行了改进,提出了一种新的基于群体适应度方差对  $gBest$  进行自适应变异的粒子

群优化算法—AMPSO,其算法流程如下:

- (1) 随机初始化粒子群中粒子的位置与速度.
- (2) 将粒子的  $pBest$  设置为当前位置,  $gBest$  设置为初始群体中最佳粒子的位置.
- (3) 判断算法收敛准则是否满足,如果满足,转向(9);否则,执行(4).
- (4) 对于粒子群中的所有粒子,执行如下操作:  
根据式(1)、(2)和式(3)更新粒子的位置与速度.  
如果粒子适应度优于  $pBest$  的适应度,  $pBest$  设置为新位置.  
如果粒子适应度优于  $gBest$  的适应度,  $gBest$  设置为新位置.
- (5) 根据式(4)与式(5)计算群体适应度方差  $\sigma^2$ ,并计算  $f(gBest)$ .
- (6) 根据式(16)计算变异概率  $p_m$ .
- (7) 产生随机数  $r \in [0, 1]$ ,如果  $r < p_m$ ,按式(17)执行变异操作;否则,转向(8).
- (8) 判断算法收敛准则是否满足,如果满足,执行(9);否则,转向(4).
- (9) 输出  $gBest$ ,算法运行结束.

从上述流程可以看出,自适应变异的粒子群优化算法实际上是在粒子群优化算法的基本框架中增加了随机变异算子,通过对  $gBest$  的随机变异来提高粒子群优化算法跳出局部最优解的能力.

### 4 自适应变异的粒子群优化算法性能分析

下面将通过四个典型函数优化问题(求解最小值)来测试本文算法的性能,同时与实数遗传算法(RGA)<sup>[6]</sup>和粒子群优化算法进行了比较.函数  $f_1(x)$  是单峰二次函数;函数  $f_2(x)$  是具有强烈振荡的多峰函数,一般算法难以得到最优解<sup>[6]</sup>;函数  $f_3(x)$  是很难极小化的病态二次函数<sup>[7]</sup>;函数  $f_4(x)$  是具有大量局部最优点的多峰函数<sup>[7]</sup>.实验设置的参数如下:三种算法的群体大小都为 30;实数遗传算法的交叉概率为 0.8,变异

概率为 0.05;粒子群优化算法与 AMPSO 算法的最大加权系数与最小加权系数都取 0.9 和 0.1.对于  $f_1(x)$ ,将  $f_1(x) + 10^{-10}$  作为适应度函数;对于  $f_2(x)$  和  $f_3(x)$ ,将函数表达式作为适应度函数;对于  $f_4(x)$ ,将  $f_4(x) + 0.1$  作为适应度函数.表 1 列出了用三种算法求解上述优化问题运行 20 次后得到的平均函数最优解以及 AMPSO 算法求解这些问题时  $k$ 、 $\frac{2}{d}$  及  $f_d$  的取值.

$$f_1(x) = \sum_{i=1}^{10} x_i^2, -100 \leq x_i \leq 100 \quad (18)$$

$$f_2(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{|1 + 0.001 * (x_1^2 + x_2^2)|^2 + 0.5}, -2 \leq x_i \leq 2 \quad (19)$$

$$f_3(x) = \sum_{i=1}^9 [100 * (x_{i+1} - x_i)^2 + (x_i - 1)^2], -100 \leq x_i \leq 100 \quad (20)$$

$$f_4(x) = \sum_{i=1}^{10} [x_i^2 - 10\cos(2\pi x_i) + 10], -100 \leq x_i \leq 100 \quad (21)$$

从表 1 可以看出:对于所有测试函数,本文算法的优化结果都明显好于其它两种算法,其中对于  $f_1(x)$ 、 $f_2(x)$  和  $f_4(x)$ ,本文算法获得了理论最优值.图 3 至图 6 是上述四个函数采用三种算法求解运行 20 次后得到的平均最佳适应度进化曲线.为了便于比较,图 3、图 5 与图 6 的纵坐标都采用适应度的对数值表示.从图中可以看出:对于单峰函数  $f_1(x)$ ,本文算法的全局收敛速度快于粒子群优化算法与实数遗传算法;粒子群优化算法在  $f_2(x)$  与  $f_3(x)$  的优化过程中都陷入局部最优,实数遗传算法在函数  $f_4(x)$  中也出现了早熟收敛情况;本文算法在所有函数优化问题中,都具有较快的全局收敛速度与强大的全局搜索能力,能有效地避免遗传算法和粒子群优化算法的早熟收敛问题.

表 1 三种算法运行 20 次的函数平均最优解

函 数	理论最优解	RGA	PSO	AMPSO
$f_1(x)$	0	5.590376E-4	8.843129E-8	0 ( $f_d=0, \frac{2}{d}=0.001, k=0.3$ )
$f_2(x)$	-1	-0.985610	-0.989869	-1 ( $f_d=-1, \frac{2}{d}=0.001, k=0.3$ )
$f_3(x)$	0	75.146586	4.022082E+5	8.660438 ( $f_d=0, \frac{2}{d}=1.1, k=0.3$ )
$f_4(x)$	0	32.088741	9.136798	0 ( $f_d=0, \frac{2}{d}=1.5, k=0.3$ )

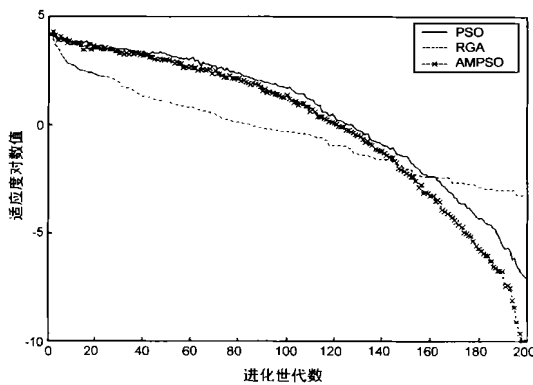


图 3  $f_1(x)$  20 次平均最佳适应度进化曲线

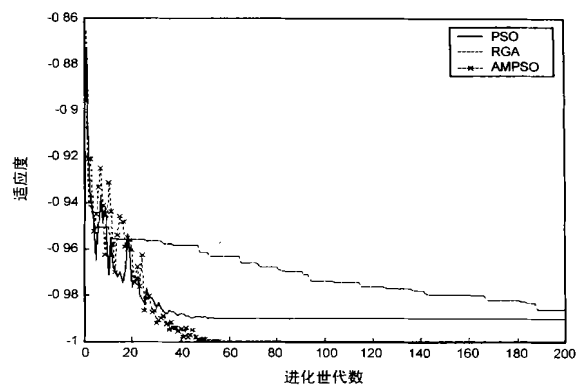


图 4  $f_2(x)$  20 次平均最佳适应度进化曲线

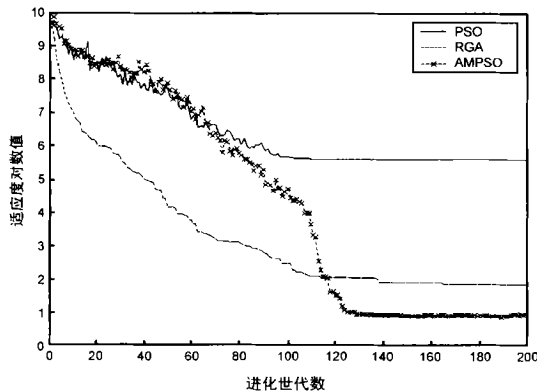


图 5  $f_3(x)$  20 次平均最佳适应度进化曲线

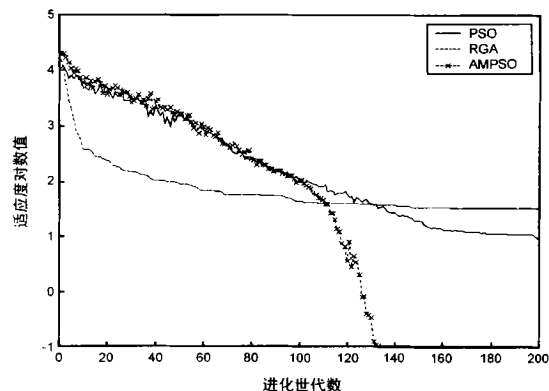


图 6  $f_4(x)$  20 次平均最佳适应度进化曲线

## 5 小结

本文针对粒子群优化算法的早熟收敛问题,提出了一种采用基于自适应变异机制的粒子群优化算法.实验表明,新算法不仅具有很强的全局搜索能力,而且能有效避免粒子群优化算法和遗传算法的早熟收敛问题.本文算法的运算量比粒子群优化算法略微有所增加,但是比遗传算法的运算量要小得多,是一个非常实用的优化工具.

### 参考文献:

- [ 1 ] J Kennedy ,R C Eberhart . Particle swarm optimization[A]. Proc IEEE international conference on Neural Networks [ C]. USA : IEEE Press , 1995 ,4. 1942 - 1948.
- [ 2 ] Y Shi ,R C Eberhart . A modified swarm optimizer[A]. IEEE International Conference of Evolutionary Computation[C]. Anchorage ,Alaska : IEEE Press ,May ,1998.
- [ 3 ] Mark S Voss ,Xin Feng . ARMA model selection using particle swarm optimization and AIC criteria[A]. 15<sup>th</sup> Triennial World Congress[C]. Barcelona ,Spain : IFAC ,2002.
- [ 4 ] F van den Bergh ,A P Engelbrecht . Cooperative learning in neural networks using particle swarm optimizers [J]. South African Computer Journal ,2000(11) :84 - 90.
- [ 5 ] F van den Bergh ,An analysis of particle swarm optimizers[D]. South Africa :Department of Computer Science ,University of Pretoria ,2002.

81 - 83.

- [ 6 ] 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社, 2001. 48 - 49.
- [ 7 ] 王小平,曹立明. 遗传算法—理论、算法与软件实现[M]. 陕西西安:西安交通大学出版社,2002. 105 - 107.

### 作者简介:



吕振肃 男,1946 年生于山西省沁水县,兰州大学信息科学与工程学院教授,研究方向为数字信号处理、智能控制、计算机网络技术.



侯志荣 男,1978 年生于四川省营山县,兰州大学信息科学与工程学院硕士研究生,研究方向为智能优化算法、数字信号处理、Internet 技术.