

文章编号: 1001-0920(2009)08-1257-04

# 一种基于种群速度的自适应粒子群算法

张顶学, 廖锐全

(长江大学 石油工程学院, 湖北 荆州 434023)

**摘要:** 分析了粒子群算法的收敛性, 指出早熟是由于粒子速度降低而失去继续搜索可行解的能力. 进而提出一种基于种群速度动态改变惯性权重的粒子群算法, 该算法以种群粒子平均速度为信息动态改变惯性权重, 避免了粒子速度过早接近 0. 通过 5 个标准测试函数的仿真实验并与其他算法相比, 结果表明该算法在进化中期能很好地保持种群多样性, 有效地改善算法的平均最优值和成功率.

**关键词:** 粒子群优化算法; 惯性权重; 种群多样性; 收敛性

**中图分类号:** TP18      **文献标识码:** A

## Adaptive particle swarm optimization algorithm based on population velocity

ZHANG Ding-xue, LIAO Rui-quan

(Petroleum Engineering College, Yangtze University, Jingzhou 434023, China. Correspondent: ZHANG Ding-xue, E-mail: zdx7773@163.com)

**Abstract:** The convergence of particle swarm optimization (PSO) algorithm is analyzed. Its premature convergence is due to the decrease of the velocity of particles in search space. An adaptive PSO algorithm with dynamical changing inertia weight based on population velocity is proposed. The information defined as the average absolute value of velocity of all particles is defined as information to change the inertia weight dynamically, which can avoid the velocity closed to 0 in the early search part. The simulation results show that the algorithm has better probability of finding global optimum and mean best value and can maintain the population diversity in the process of evolution.

**Key words:** Particle swarm optimization algorithm; Inertia weight; Population diversity; Convergence

### 1 引言

粒子群优化(PSO)算法是一种新的基于群体智能的优化算法<sup>[1,2]</sup>. 由于 PSO 算法结构简单, 参数调整简便易行, 适合计算机编程处理, 一经提出便受到广泛的关注, 目前已成功地应用于很多领域.

PSO 算法中惯性权重对算法的性能有很大的影响. 基本 PSO 算法中采用固定惯性权重, 在复杂问题求解时, 极易陷入局部最优而早熟收敛. 为此, 学者们提出了多种惯性权重的改变策略<sup>[2-7]</sup>, 在一定程度上改进了 PSO 算法的性能.

本文在分析 PSO 算法中种群速度对算法性能影响的基础上, 提出一种动态改变惯性权重的自适应粒子群算法. 仿真实验结果表明, 该算法在进化中期能很好地保持种群多样性, 有效地改善算法的平均最优值和成功率.

### 2 PSO 算法

PSO 算法是模拟鸟群飞行觅食的行为, 通过个体之间的协作来寻找最优解的进化计算技术. 假设其搜索空间为  $n$  维, 粒子总数为  $m$ , 第  $i$  个粒子在  $n$  维空间的位置为  $x_i$ , 飞行速度为  $v_i$ . 每个粒子都有一个由被优化的目标函数决定的适应值, 并且知道自己目前为止所发现的最好位置  $p_i$  和现在的位置, 都知道目前为止整个群体所发现的最好位置  $p_g$ . 则每个粒子的位置按下式进行更新<sup>[1,2]</sup>:

$$v_i^{(k+1)} = \omega v_i^{(k)} + c_1 r_1 (p_i - x_i^{(k)}) + c_2 r_2 (p_g - x_i^{(k)}), \quad (1)$$

$$x_i^{(k+1)} = x_i^{(k)} + v_i^{(k+1)}. \quad (2)$$

其中:  $v_i^{(k)}$  为第  $i$  个粒子第  $k$  次迭代的飞行速度,  $x_i^{(k)}$  为第  $i$  个粒子第  $k$  次迭代的位置,  $r_1$  和  $r_2$  为  $[0, 1]$  内的随机数,  $c_1$  和  $c_2$  为权重因子,  $\omega$  为惯性权重. 研究

收稿日期: 2008-09-22; 修回日期: 2008-12-30.

基金项目: 国家自然科学基金项目(60873021); 湖北省高等学校优秀中青年团队计划项目(T200803).

作者简介: 张顶学(1975—), 男, 湖北潜江人, 讲师, 博士, 从事智能计算、智能控制的研究; 廖锐全(1962—), 男, 广东梅州人, 教授, 博士生导师, 从事复杂系统理论及其应用等研究.

表明,较大的惯性权重有利于全局探索;较小的惯性权重有利于算法的局部开发,加速算法的收敛<sup>[2]</sup>.

为了提高算法的性能,许多文献对惯性权重进行改进.文献[2]提出随着进化代数的增加而线性减小惯性权重的策略(LWPSO),该策略能在进化初期加强全局探索能力,在进化后期加强算法的局部开发能力,使算法的性能得到提高,但在复杂问题求解时效果不很明显.[3]提出用模糊控制器来自适应改变惯性权重,但算法需要建立模糊规则,其实现比较复杂.[4]提出了随机惯性权重策略(RWPSO),仿真结果表明,该进化策略对于大多数多峰函数能快速收敛,并能发现较好的解.[5]提出一种混沌惯性权重改变策略(CPSO),其惯性权重按下式改变:

$$\begin{aligned} \omega(t+1) &= 4.0\omega(t)(1-\omega(t)), \\ \omega(t) &\in (0,1). \end{aligned} \quad (3)$$

文献[6]提出一种按指数形式递减的惯性权重改变策略,分别给出了如下两种变化形式:

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \exp(-10 \times t / \max \text{ter}), \quad (4)$$

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \exp[-(t / (\max \text{ter} / 4))^2]. \quad (5)$$

这里分别称为 EPSO1 和 EPSO2.其中:  $\omega_{\max}$  和  $\omega_{\min}$  分别为  $\omega$  变化的最大值和最小值,  $\max \text{ter}$  为最大进化代数.

文献[7]提出一种动态改变的惯性权重策略(IPS0),其惯性权重的改变形式为

$$\begin{aligned} \omega' &= \omega u^{-t}, \omega \in [0,1], \\ u &\in [1.0001, 1.005]. \end{aligned}$$

并通过仿真实验得到了  $u$  的最优取值,但该算法在种群较小的情况下不能取得较好的效果.

### 3 基于种群速度的自适应粒子群算法

#### 3.1 PSO 算法的收敛性分析

文献[8-10]分析了 PSO 算法中粒子轨迹的收敛性条件,在此基础上分析了粒子速度对算法收敛性的影响.设

$$\varphi_1 = c_1 r_1, \varphi_2 = c_2 r_2, \varphi = \varphi_1 + \varphi_2.$$

由式(1)和(2)消去速度相关项,可得

$$\begin{aligned} x_i^{(k+1)} &= (1 + \omega - \varphi)x_i^{(k)} - \\ &\quad \omega x_i^{(k-1)} + \varphi_1 p_i + \varphi_2 p_g. \end{aligned} \quad (6)$$

式(6)的齐次矩阵形式为

$$[x_i^{(k+1)} \quad x_i^{(k)} \quad 1]^T = A[x_i^{(k)} \quad x_i^{(k-1)} \quad 1]^T, \quad (7)$$

其中

$$A = \begin{bmatrix} 1 + \omega - \varphi & -\omega & \varphi_1 p_i + \varphi_2 p_g \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

式(7)中系数矩阵的特征多项式为

$$(\lambda - 1)(\lambda^2 - (1 + \omega - \varphi)\lambda + \omega) = 0. \quad (8)$$

方程(8)存在 3 个特征根,即

$$\begin{aligned} \lambda_1 &= 1, \\ \lambda_2, \lambda_3 &= (1 + \omega - \varphi \pm \\ &\quad \sqrt{(1 + \omega - \varphi)^2 - 4\omega}) / 2. \end{aligned}$$

由此式(6)可写成

$$x_i^{(k)} = p_1 + p_2 \lambda_2^k + p_3 \lambda_3^k.$$

由式(1)和(2)消去位置项,可得

$$v_i^{(k+1)} - (1 + \omega - \varphi)v_i^{(k)} + \omega v_i^{(k-1)} = 0, \quad (9)$$

其中系数矩阵特征多项式为

$$\lambda^2 - (1 + \omega - \varphi)\lambda + \omega = 0. \quad (10)$$

由于  $\lambda_2$  和  $\lambda_3$  为方程(10)的两个特征根,可得

$$v_i^{(k)} = q_1 \lambda_2^k + q_2 \lambda_3^k. \quad (11)$$

文献[9]已证明,保证粒子轨迹收敛的条件是当且仅当  $\max(\|\lambda_2\|, \|\lambda_3\|) \leq 1$ . 在粒子轨迹收敛的情况下,分析其速度收敛情况,即  $\lim_{k \rightarrow \infty} v_i^{(k)} = \lim_{k \rightarrow \infty} (q_1 \lambda_2^k + q_2 \lambda_3^k)$ . 可分两种情况讨论:

1) 当  $\max(\|\lambda_2\|, \|\lambda_3\|) < 1$  时,  $\lim_{k \rightarrow \infty} v_i^{(k)} = 0$ ;

2) 当  $\max(\|\lambda_2\|, \|\lambda_3\|) = 1$  时,粒子速度收敛,且  $\lim_{k \rightarrow \infty} v_i^{(k)} = q_1$  或  $q_2$  或  $q_1 + q_2$ .

在 2) 的情况下,粒子的速度始终以某一速度运行,粒子的探索能力较强,但局部开发能力较弱,在一定的进化代数内很难搜索到较好的解;在 1) 的情况下,粒子的速度会逐渐变小直至等于 0,在进化过程中,若粒子没有搜索到全局最优而粒子速度提前达到 0 或非常接近 0,则优化过程会出现停滞,从而导致早熟的发生.

为了避免早熟,提高算法的适应性能,本文根据种群粒子速度来动态改变惯性权重,从而调整粒子速度的大小,避免粒子速度提前达到 0 或接近 0.

#### 3.2 种群粒子的平均速度

粒子速度有正有负,无论粒子速度是正是负,粒子位置在进化的下一代都会发生变化,因此种群粒子的平均速度可由种群中各粒子的平均绝对速度来表示.种群粒子在第  $k$  代的平均速度  $v_{\text{avg}}^{(k)}$  表示为

$$v_{\text{avg}}^{(k)} = \left( \sum_{i=1}^m \sum_{j=1}^n |v_{ij}^{(k)}| \right) / (nm). \quad (12)$$

平均速度  $v_{\text{avg}}^{(k)}$  的大小反映了粒子搜索空间的大小,较大的速度,种群粒子的搜索空间较大,粒子的探索能力较强,开发能力较弱;较小的速度,种群粒子在较小的空间搜索,粒子的开发能力较强,探索能力较弱.在群智能优化算法中,早期搜索阶段希望个体能搜索整个解空间,不要在局部最优附近聚集;在进化

后期主要是加强全局最优解的收敛,使算法能有效发现最优解.

### 3.3 惯性权重的改变策略

本文根据种群速度的大小来调整惯性权重的大小,从而使种群速度按照期望的方式下降,防止种群速度过早等于 0 或接近 0. 当种群速度大于期望速度时,应减小惯性权重;反之,则应增大惯性权重. 假设初始的惯性权重为  $w_0$ , 第  $k$  代种群粒子平均速度的期望值为  $v_e^{(k)}$ . 实际种群的粒子速度为  $v_{avg}^{(k)}$ , 第  $k$  代惯性权重为  $w(k)$ ,  $p$  为常数. 则第  $k+1$  代惯性权重按以下方式改变:

- If  $v_{avg}^{(k)} > v_e^{(k)}$ , Then  $w(k+1) = w(k)/p$ ;
- If  $v_{avg}^{(k)} < v_e^{(k)}$ , Then  $w(k+1) = w(k) \times p$ ;
- If  $v_{avg}^{(k)} = v_e^{(k)}$ , Then  $w(k+1) = w(k)$ .

当  $w(k+1) > w_{max}$  时,取  $w(k+1) = w_{max}$ ; 当  $w(k+1) < w_{min}$  时,取  $w(k+1) = w_{min}$ . 进化过程中种群平均速度必然减小直至为 0.

在进化初期,为了加强算法的探索能力,应保持较大的种群平均速度;在进化后期,为了加强算法的开发能力,应保持较小的种群平均速度. 设初始种群的平均速度为  $v_0$ , 最大进化代数数为  $T_{max}$ . 参照模拟退火算法的退火函数,期望种群速度采用指数形式变化,即

$$v_e^{(k)} = v_0 e^{-(2k/(T_{max}-T_1))^2}. \quad (13)$$

## 4 仿真实验

为了测试本文提出的粒子群算法(VWPSO),对 5 个标准测试函数进行仿真实验,测试函数如表 1 所示. 将本文算法与其他算法相比较来分析算法性能. 对每种算法采用两类实验模式:模式 I:给定最

表 1 测试函数的维数、搜索空间和目标最优值

	函 数	维数	搜索空间	目标最优值
$f_1$	Sphere	30	$[-100,100]^{30}$	0.01
$f_2$	Rosenbrock	30	$[-100,100]^{30}$	100
$f_3$	Rastrigrin	30	$[-100,100]^{30}$	100
$f_4$	Griewank	30	$[-600,600]^{30}$	0.1
$f_5$	Ackley	30	$[-32,32]^{30}$	0.1

大进化代数  $T_{max}$ , 执行测试算法;模式 II:给定最大进化代数  $T_{max}$  和目标最优值  $f_b$ , 执行测试算法.

为了评价算法的优劣,采用以下两个评价指标:平均最优适应值  $f_{avg}$ , 即对模式 I 进行  $M$  次实验所得函数值的算术平均值;平均成功率  $R$ , 即对模式 II 进行  $M$  次实验,其中  $M_s$  次实验搜索到的最优值小于目标最优值  $f_b$ , 平均成功率  $R = M_s/M \times 100\%$ .

各算法的粒子种群数为 20,  $w_{max} = 0.95$ ,  $w_{min} = 0.4$ , 最大进化代数均为  $T_{max} = 2000$ , 每种算法独立运行 10 次. 在 VWPSO 算法中,局部开发代数  $T_1 = 400$ , 初始惯性权重  $w_0 = 0.95$ ,  $p$  由实验确定.

### 4.1 $p$ 值的确定

$p$  为惯性权重的变化率,较大的取值将使惯性权重变化较大,从而使种群速度变化较大,能增强种群的多样性,但算法的局部开发能力减弱;较小的取值将使算法的局部开发能力增强,但种群的多样性有可能减弱. 为了确定合适的  $p$  值,选取 4 个不同的值 1.03, 1.04, 1.05 和 1.06 进行仿真实验,其结果如表 2 所示.

从表 2 可以看出,  $p = 1.05$  时 VWPSO 能取得

表 2 不同  $p$  值的仿真结果

函数	1.03		1.04		1.05		1.06	
	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$
$f_1$	$2.21 \times 10^{-9}$	100	$4.42 \times 10^{-10}$	100	$2.08 \times 10^{-10}$	100	$1.06 \times 10^{-9}$	100
$f_2$	101.7802	50	117.4821	90	73.622	90	157.6959	70
$f_3$	71.1399	100	65.7669	100	65.171	100	65.9656	100
$f_4$	0.02159	100	0.00665	100	0.01524	100	0.01378	100
$f_5$	$9.53 \times 10^{-6}$	100	0.1778	90	$2.44 \times 10^{-10}$	100	$4.53 \times 10^{-6}$	100

表 3 各种算法计算结果对比

函数	VWPSO		LWPSO		RWPSO		CPSO		EPSO1		EPSO2		IPSO	
	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$	$f_{avg}$	$R$
$f_1$	2.08E-10	100	3.36E-8	100	126.307	0	1.50E-24	100	2.28E-25	100	1.58E-20	100	<b>4.15E-35</b>	100
$f_2$	73.622	90	241.420	40	107.211	60	174.230	60	57.755	80	63.831	80	<b>35.36</b>	100
$f_3$	<b>65.171</b>	100	98.102	60	85.168	70	72.930	90	82.879	90	77.209	90	140.98	10
$f_4$	<b>0.01524</b>	100	0.01721	100	0.02085	90	0.01723	100	0.02207	100	0.01746	100	0.0174	100
$f_5$	<b>2.44E-10</b>	100	0.3198	70	3.5926	0	1.28E-5	100	0.116	90	0.523	60	2.4895	10



图1 5个函数进化过程中的平均适应度变化

较高的成功率,除  $f_4$  和  $f_5$  所得的平均最优值与最小平均最优值相差无几外,其余3个函数的平均最优值都最小.综合考虑成功率和平均最优值,取  $p = 1.05$ .

### 4.2 与其他算法的比较

VWPSO 与 LWPSO, RWPSO, CPSO, EPSO1, EPSO2 和 IPSO 算法的对比结果如表3所示.从表中可以看出,对于3个多峰函数,无论是平均最优值还是成功率, VWPSO 算法都有所改善.对于单峰函数  $f_1$  和单峰病态函数  $f_2$ , VWPSO 算法却很难显示出其优势.

各函数进化过程中的平均适应度变化如图1所示.可以看出,对于3个多峰函数, VWPSO 算法在早中期并没有优势,在进化后期能较好地改善适应度.

$f_3$  进化过程中的平均种群速度和惯性权重的变化如图2和图3所示.从图2可以看出: RWPSO 和 IPSO 算法种群速度下降最快,在700代左右已接近  $10^{-4}$ ,因此中后期其适应度基本没有发生变化. VWPSO 算法在进化的早期和中期保持较大的种群

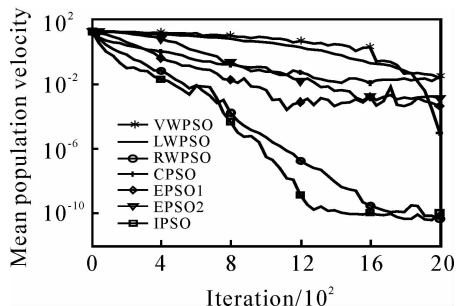


图2 种群速度变化

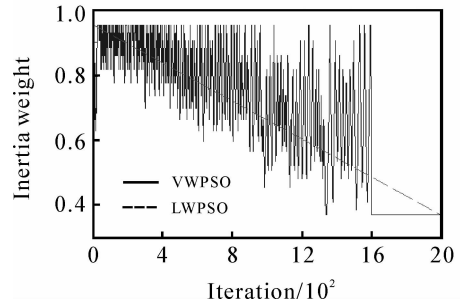


图3 惯性权重变化

速度,从而增强了算法的全局探索能力,具有较高的种群多样性,在进化的后期其种群速度迅速下降,增强了算法的局部开发能力,因此算法取得了较好的效果.

从图3可以看出, VWPSO 算法的惯性权重在进化过程中是动态变化的.正是由于其惯性权重的动态改变,平衡了算法的全局探索和局部开发能力,保持了种群的多样性,不仅能避免陷入局部最优,而且可得到较好的最优值.

## 5 结论

本文分析粒子群算法在轨迹收敛条件下粒子速度的收敛情况,以及粒子速度大小的变化对算法性能的影响,提出用种群粒子的平均绝对速度来动态改变惯性权重,由此引导种群速度的变化,避免进化过程中过早接近于0而出现早熟.通过对5个标准测试函数的测试并与其他5种粒子群算法相比较,结果表明该算法对于多峰函数在成功率上和平均最优值上都有很大的改善,但对于单峰函数效果则不明显.

(下转第1265页)