

文章编号: 1001-0920(2010)10-1463-06

## 拥有领导机制的改进粒子群算法

周龙甫<sup>1,2</sup>, 师奕兵<sup>2</sup>, 张 伟<sup>2</sup>

(1. 解放军第 452 医院 信息科, 成都 610061; 2. 电子科技大学 自动化工程学院, 成都 611731)

**摘要:** 为了提高粒子群算法的全局收敛能力和收敛速度, 在以往文献的基础上提出一种改进粒子群算法. 受生物学研究成果的启发, 引入领导机制, 将粒子群搜索过程分为领导粒子带领下的探索性搜索和所有粒子共同参与的开发性搜索两部分. 通过“变异”机制来增强群体多样性, 采用 5 个标准函数对算法性能进行分析, 应用 Bonferroni 多重比较法, 将改进算法与两种经典算法进行基本性能对比. 实验结果表明, 所提出的改进算法探索速度快, 全局搜索能力优.

**关键词:** 粒子群; 领导机制; 变异; 方差分析

**中图分类号:** TP273

**文献标识码:** A

## Improved particle swarm optimization with leadership

ZHOU Long-fu<sup>1,2</sup>, SHI Yi-bing<sup>2</sup>, ZHANG Wei<sup>2</sup>

(1. Department of Information, The 452nd Hospital of PLA, Chengdu 610061, China; 2. College of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. Correspondent: ZHOU Long-fu, E-mail: zhoulf\_1977@163.com)

**Abstract:** In order to enhance the global convergence ability and the convergence speed of particle swarm optimization (PSO), based on the results of reference, an improved strategy is proposed. Inspired by the effect of individual experience in animal groups on the move, the whole searching process is departed into two steps, the exploration search with leadership and the exploitation search. The mutation operator is used to add the diversity of swarm. By analyzing five benchmark functions, the improved method is compared with two famous PSO methods. And the variance analysis of statistic theory is applied to compare the performance of the three methods. Experimental simulation results show that the proposed algorithm can not only significantly speed up the convergence, but also effectively solve the premature convergence problem.

**Key words:** Particle swarm; Leadership; Mutation; Statistic analysis

## 1 引 言

1995 年, Eberhart 和 Kennedy 通过对鸟群捕食行为的研究, 提出了一种计算智能方法——粒子群优化算法 (PSO)<sup>[1]</sup>. 随后, 人们引入了惯性权重的概念, 以实现了解空间搜索更好的控制, 逐步形成了目前普遍应用的基本微粒群算法<sup>[2]</sup>. PSO 算法简单、便于实现, 能有效解决复杂优化任务<sup>[3]</sup>.

对于 PSO 算法本身的理论研究包括单个粒子飞行轨迹分析和整个群体行为研究, 即微观分析和宏观研究两个层面.

早在 1997 年, Kennedy<sup>[4]</sup>便提到了单个粒子的轨迹依赖于随机参数的取值, 而且随机参数的取值将影响整个粒子群的搜索能力. 但是, 这些参数如何影响

单粒子轨迹和粒子群搜索能力的机制尚不明确. 由此拉开了对 PSO 算法微观层面上各个参数对粒子轨迹收敛性研究的序幕. 在该方面, 首次理论分析的尝试<sup>[5]</sup>便发现, 在一定条件下一维空间中单粒子位置轨迹按正弦规律变化. 文献 [6,7] 忽略了随机数的影响, 将粒子运动状态简化为线性, 重点关注惯性权重、学习因子对粒子轨迹的作用. 随后, 人们将非线性反馈理论和 Mason 信号流图理论分别引入随机量对粒子轨迹稳定收敛影响的研究<sup>[8,9]</sup>, 得到了与 [6,7] 相似的惯性权重与随机参数取值条件. 在宏观层面上对基本 PSO 算法的改进所得到的研究成果层出不穷, 其中将遗传算法的“变异”操作引入 PSO 算法是一种增强全局搜索性能的有效方法. [10] 通过对微粒速度或

收稿日期: 2009-09-15; 修回日期: 2009-11-18.

基金项目: 教育部新世纪人才支持计划项目 (NCET-05-0804); 国家 863 计划项目 (2006AA06Z222).

作者简介: 周龙甫 (1977—), 男, 成都人, 博士生, 从事人工智能、模拟电路智能诊断的研究; 师奕兵 (1964—), 男, 成都人, 教授, 博士生导师, 从事 VLSI 和电子系统的计算机辅助测试等研究.

位置引入一个小概率随机变异操作来增强种群多样性;在自组织算法的基础上,[11]给出了一种变异操作随时间变化的自适应层次 PSO 算法,并给出了适合变异操作的自适应参数选择方式;群体适应度方差自适应变异的引入<sup>[12]</sup>,增强了粒子群优化算法跳出局部最优解的能力;结合生物群体信息扩散的习性,[13]提出了自适应扩散混合变异机制微粒群算法。

虽然 PSO 算法发展迅速,并取得了许多研究成果,但如何将单粒子运动状态的微观层面研究与算法改进相结合仍然是 PSO 算法研究领域尚待解决的问题之一。为此,本文以单粒子运动轨迹收敛的随机数取值条件为基础,借鉴生物界集群中首领对群体行为的作用,提出一种改进 PSO 算法。

## 2 PSO 算法原理

标准 PSO 算法的数学描述为:在  $D$  维搜索空间中,  $m$  个粒子组成一个群体,第  $i$  个粒子在  $D$  维空间中的位置表示为  $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ,每个粒子的飞行速度为  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD}), i = 1, 2, \dots, m$ 。第  $i$  个粒子经历过的个体最好位置  $P_{best}$  (有个体最佳适应度值)记为  $p_b$ ,整个群体中所有粒子的全局最佳位置  $g_{best}$  (有群体最佳适应度值)记为  $g_b$ 。每一代粒子的速度和位置迭代方程如下:

$$\begin{aligned} v(k+1) = & \\ \omega v(k) + \varphi_1(k)(p_b(k) - x(k)) + & \\ \varphi_2(k)(g_b(k) - x(k)), & \quad (1) \end{aligned}$$

$$x(k+1) = x(k) + v(k+1). \quad (2)$$

其中:  $\omega$  为惯性权重;  $\varphi_1 = c_1 * \text{rand}(0, 1), \varphi_2 = c_2 * \text{Rand}(0, 1), c_1$  和  $c_2$  是学习因子,在本文中设定为常数 1.5;  $\text{rand}(0, 1)$  和  $\text{Rand}(0, 1)$  表示区间  $[0, 1]$  上的两个相互独立的随机数。

在 PSO 算法研究过程中,对基本 PSO 算法有 2 种重要的改进:

1) 自适应时变权重 PSO 算法。文献[14]提出了惯性权重  $\omega$  随迭代次数线性变化,从而增强了粒子群全局探索能力和局部开发能力。本文记为 PSO-TVIW。

2) 固定权重 PSO 算法。为了提高 PSO 算法对动态问题的搜索能力,文献[6]提出并验证了当惯性权重  $\omega = 0.729$ ,学习因子  $c_1 = c_2 = 1.494$  时,算法的搜索能力更强。本文记为 PSO-CF。

## 3 拥有领导机制的改进 PSO 算法 (PSO-L)

### 3.1 改进算法的理论基础

#### 3.1.1 算法提出的生物学基础

生物学家 Couzin 在文献[15]中指出:自然界中,对于群体觅食或迁徙的动物,通常是群体之间相互交

流信息作出行动决定。然而,在很多情况下,一些个体拥有相关信息,能够有效带领集群觅食或迁徙。它们甚至没必要向周围的动物发出特殊的信号,就可以在集群移动时制造出一种倾向,让整个集群向特定的方向前进。

Couzin 通过分析和研究发现,群体数目越大,这种带头个体的数目越小,而且很少的带头个体可以让整个群体的行动获得更大的准确性。

#### 3.1.2 单粒子运动轨迹稳定条件

考虑到随机数取值的影响,文献[16]将单个粒子在随机量影响下的运动方程转换为二阶变系数非齐次方程,应用 Lyapunov 稳定定理对该方程的稳定性进行研究,分析得到了使粒子运动稳定收敛的惯性权重和随机参数取值条件:

**条件 1**  $0 < \omega < 1$ ;

**条件 2**  $0 < \varphi_1(k+1) + \varphi_2(k+1) < 2 + 2\omega$ ;

**条件 3**  $|\varphi_1(k+1) - \varphi_1(k) + \varphi_2(k+1) - \varphi_2(k)| < \frac{1}{10}(1 - \omega)(\varphi_1(k+1) + \varphi_2(k+1))(2 + 2\omega - \varphi_1(k+1) - \varphi_2(k+1))$ 。

将上述条件与文献[6,7]所得的最终结果相比较,条件 1 与[6,7]所得的最终结果完全一致;[6,7]中关于惯性权重  $\omega$  的取值条件是当随机量取值不变情况下条件 2 的一个特例。[16]和[6,7]从不同的分析角度相互验证了所得成果的正确性。为了使粒子运动轨迹稳定,条件 3 进一步约束了随机量的取值。显然,PSO-CF 算法中的惯性权重和学习因子参数取值满足条件 1 和条件 2。

受生物学理论研究结果的启发,结合文献[16]的研究成果,本文提出一种拥有领导机制的改进 PSO 算法。

### 3.2 PSO-L 算法原理

群体随机优化算法中探索能力是指算法能够在搜索空间探索新的领域,而开发能力是指算法能够在某个小的区域进行彻底的搜索。对于一个好的优化算法而言,二者都非常重要,但它们之间又相互矛盾,只有很好地平衡二者之间的关系才能使算法最终收敛或收敛几率很高。本算法中,将整个粒子群的搜索过程分为探索性搜索和开发性搜索两部分。探索性搜索过程中,在领导粒子作用下,着力快速找到或靠近全局最优值点;而后在探索性搜索的基础上,通过全体粒子共同参与的开发性搜索过程进行彻底搜索。算法流程如图 1 所示。

#### 3.2.1 粒子群初始化与领导粒子确立

在很多情况下,粒子的搜索范围是关于零点对称的,如果直接在其搜索范围上对粒子进行初始化,则

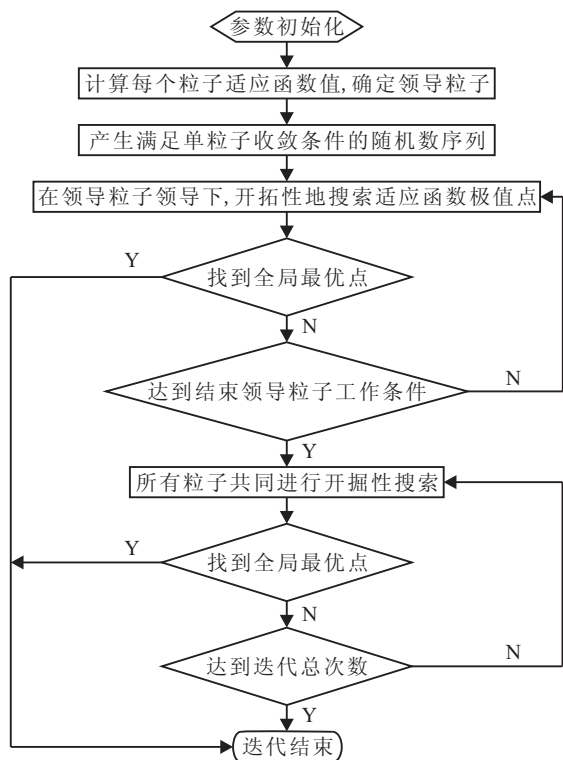


图 1 PSO-L 算法流程图

粒子比较容易收敛于最优值点. 为了避免粒子的初始化对算法造成的影响, 在 PSO 算法研究中往往在不包含零点的偏区间上对粒子进行初始化. 具体的初始化区间根据函数的搜索空间确定.

在粒子初始化后, 计算每个粒子的适应度函数值 (VFF). 按所研究问题的不同, 选择 VFF 优 (最大或最小) 的粒子作为“领导”粒子. 当出现多个粒子同时具有最优 VFF 时, 选择速度向量模最大的粒子为“领导”粒子; 当有多个粒子同时具有最优 VFF 值和最大速度向量模时, 按整个粒子群粒子总数, 参照文献 [15] 的研究结果选择一定数目的粒子作为“领导”粒子.

### 3.2.2 领导粒子作用下探索性搜索

粒子群中的粒子运动轨迹处于发散振荡状态, 显然对算法收敛无益, 因为粒子的位置将趋向无限大, 无法对有限问题空间进行有意义的搜索; 而处于幅值有限的振荡状态则对算法有非常重要的作用, 有利于粒子群在有限问题空间内进行充分的搜索. 能否通过 PSO 参数的选择来调整粒子运动轨迹的振荡幅值呢? 显然答案是肯定的.

文献 [16] 得到了使单个粒子运动轨迹收敛的参数取值条件. 如果将这些条件作为“领导”粒子的参数取值条件, 则该“领导”粒子的运动轨迹也将是收敛的.

在 PSO 算法中, 惯性权重在区间 (0,1) 上取值, 当惯性权重很大时, 粒子具有很强的探索能力; 当惯性

权重很小时, 粒子具有很强的开发能力. 模拟自然界中的领导作用, 在初始化粒子群中找到“领导”粒子后, 赋予该“领导”粒子很大的惯性权重, 如  $\omega=0.9$ . 同时, 迭代过程中随机数取值满足文献 [16] 得到的取值条件. 其他粒子作为粒子群中的普通劳动者赋予很小的惯性权重, 如  $\omega=0.3$ . 整个粒子群在“领导”粒子的作用下共同去搜索全局最优. “领导”粒子运动轨迹必然稳定收敛, 从而有可能带领整个粒子群快速地搜索到或者靠近所求解函数的全局最优.

### 3.2.3 全体粒子共同进行的开发性搜索

当领导粒子作用下的探索性搜索结束, 粒子群还没有找到适应度函数的全局最优极值点时, 群中所有粒子共同进入开发性搜索. 根据探索性搜索结束条件, 开发性搜索采用不同的惯性权重取值方法.

当探索性搜索中部分粒子已经找到全局最优极值点时, 惯性权重  $\omega$  取值按正弦递增规律取值, 算法表现出较强的局部开发能力, 经过进一步运算使全部或大部分粒子找到全局最优值点; 当探索性搜索达到设定的领导粒子迭代次数, 没有找到全局最优极值点时, 开发性搜索中惯性权重  $\omega$  取值按余弦递减规律取值, 使算法保持一定的探索能力, 所有粒子共同搜索, 寻找全局最优极值点.

以预先设定的最大迭代次数或全体粒子搜索得到的平均 VFF 阈值作为搜索结束条件. 在满足条件情况下, 粒子群找到全局最优值点, 则本次搜索成功; 否则, 搜索不成功, 重新进行下一次搜索.

### 3.2.4 粒子迭代过程中的“变异”

首先定义迭代过程中适应度变化率 (FCR)

$$FCR = \frac{Fitness(k+1) - Fitness(k)}{Fitness(k)}, \quad (3)$$

式中  $Fitness(k+1)$  是第  $k+1$  次迭代后得到的 VFF 值. PSO 算法容易陷入局部最优点, 为提高算法的全局搜索能力, 保持整个粒子群的多样性, 借鉴文献 [10-13] 将遗传算法中的“变异”理论引入算法中. 预先设立 2 个阈值: FCR 阈值和 VFF 阈值.

迭代过程中, 当粒子的 FCR 低于预先设定的阈值时, 说明粒子在局部位置作微小变化. 此时粒子有可能已经找到全局最优值点, 也有可能陷入了局部最优点. 当粒子的适应度函数值低于预设的 VFF 阈值时, 则认为该粒子已经找到了全局最优值点.

在探索性搜索过程中, 若领导粒子的 FCR 低于预先设定的阈值, 而领导粒子的 VFF 尚未达到阈值条件, 则可以判定领导粒子陷入局部最优点. 此时, 随机地选取领导粒子速度向量中的一维, 以一定的概率和步长重新初始化该维速度值. 同样地, 在开发性搜索过程中, 当粒子群中一定数量的粒子陷入局部最优

时,从粒子群中随机选取一个粒子,并随机选定速度向量中的一维,然后在该粒子速度向量上重新初始化速度值.变异过程的伪码如下:

```

begin
  if 领导粒子 FCR 低于阈值
    if 领导粒子的 VFF 低于阈值
      随机选择一维: =  $r$ 
      if  $\text{rand } 1() < P_m$ 
         $V_{r} = V_{r} + \text{rand } 3() * 0.5 * V_{\max};$ 
      else
         $V_{r} = V_{r} - \text{rand } 3() * 0.5 * V_{\max};$ 
      end if
    end if
  end if
  if 一部分粒子的 FCR 低于阈值
    if 这些粒子的 VFF 低于阈值
      随机选择一个粒子: =  $k$ 
      随机选择一维: =  $q$ 
      if  $\text{rand } 4() < P_m$ 
         $V_{kq} = V_{kq} + \text{rand } 5() * 0.5 * V_{\max};$ 
      else
         $V_{kq} = V_{kq} - \text{rand } 6() * 0.5 * V_{\max};$ 
      end if
    end if
  end if
end

```

其中:  $\text{rand}_i()$  ( $i=1, 2, \dots, 6$ ) 是在区间  $[0,1]$  上独立产生的均匀分布的随机数;  $P_m$  是变异概率;  $r, k$  和  $q$  是常数.

## 4 算法性能测试实验及结果分析

### 4.1 实验设计

为了证明算法的有效性,利用经典的 PSO-CF, PSO-TVIW 和本文提出的 PSO-L 算法进行性能对比研究.对表 1 所示的 5 个标准测试函数<sup>[17]</sup>进行求解.表 1 给出了每个测试函数的搜索空间取值范围和位置初始化取值范围.

表 1 所示 ( $D$  表示搜索空间维数)的 5 个测试函数中,  $f_1 \sim f_3$  是整个搜索空间上的单峰函数.其中:  $f_1$  和  $f_2$  的最优状态和最优值为  $\min f(x^*) = f(0, \dots, 0) = 0$ ;  $f_3$  的最优状态和最优值为  $\min f(x^*) = f(1, 1) = 0$ ;  $f_4$  和  $f_5$  是整个搜索空间上的多峰函数,最优状态和最优值为  $\min f(x^*) = f(0, \dots, 0) = 0$ .函数在全局最优解附近存在无穷多个局部极小将其包围,并且强烈振荡.

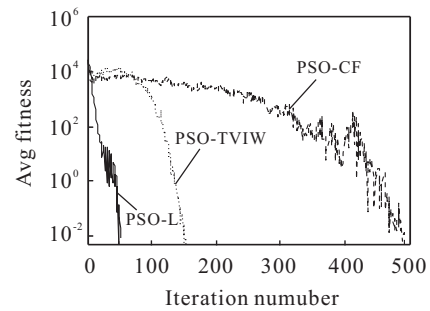
算法性能测试指标如下:AVG 为迭代结束后全局最优解均值;SR 为有限迭代次数中算法成功率;ITL 为达到判定阈值的算法迭代次数;PCL 为迭代

结束后,达到全局最优点的粒子个数.

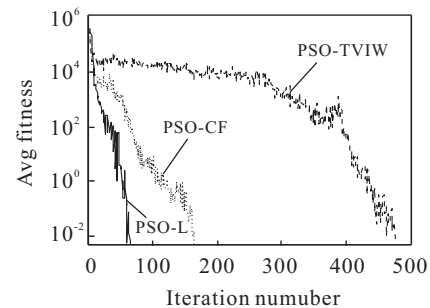
### 4.2 实验结果及统计分析

每次实验的最大迭代次数设定为 1000 次.在粒子总数、搜索空间维数不同的组合情况下,每种算法对每个函数独立运行 50 次.VFF 阈值设定为 0.05, FCR 阈值设定为 0.1.测试结果如表 2 所示.其中当函数搜索空间为 10 维时,由于 PSO-CF 和 PSO-TIVW 搜索成功率为 0,表 2 中没有给出函数搜索空间为 10 维时 PSO-CF 和 PSO-TIVW 的其他测试指标.

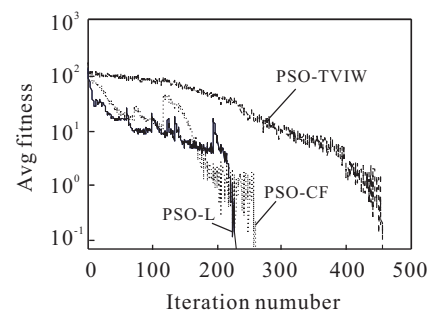
图 2 对 PSO-CF, PSO-TVIW 和 PSO-L 算法的粒子轨迹稳定收敛程度进行了例证.图 2 所示例子中,



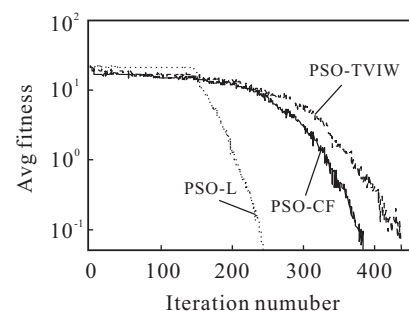
(a) Sphere 函数粒子运动轨迹



(b) Schwefell 函数粒子运动轨迹



(c) Rastrigin 函数粒子运动轨迹



(d) Ackey 函数粒子运动轨迹

图 2 达到收敛条件所需迭代次数示例图

表 1 标准测试函数

函数名	数学表达式	搜索范围	初始化范围	$V_{max}$
$f_1$ Sphere	$f_1 = \sum_{i=1}^n x_i^2$	$(-100, 100)^D$	(50,100)	100
$f_2$ Schwefel 1.2	$f_2 = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$	$(-100, 100)^D$	(50,100)	100
$f_3$ Rosenbrock	$f_3 = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$	$(-30, 30)^2$	(15,30)	30
$f_4$ Rastrigin	$f_4 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$(-5.12, 5.12)^D$	(2.56,5.12)	5.12
$f_5$ Ackey F1	$f_5 = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$(-32, 32)^D$	(16,32)	32

表 2 50 次随机运行后, 3 种算法性能测试结果

函数	维数	粒子数	AGV			SR/%			ITL			PCL		
			PSO-CF	PSO-TVIW	PSO-L	PSO-CF	PSO-TVIW	PSO-L	PSO-CF	PSO-TVIW	PSO-L	PSO-CF	PSO-TVIW	PSO-L
$f_1$	5	30	2.93E-06	1.03E-07	7.81E-05	100	100	100	119.52	444.86	45.68	16.26	21.16	8.52
	5	50	4.89E-07	9.91E-09	1.07E-04	100	100	100	124.54	443.46	37.74	31.84	37.14	14.30
	10	30	1.14E-04	4.75E-05	2.68E-03	100	100	100	157.24	475.34	161.80	8.48	13.72	15.46
$f_2$	5	30	3.47E-06	8.98E-07	3.01E-05	100	100	100	159.20	472.90	175.00	18.22	22.14	20.42
	5	50	6.17E-07	7.01E-08	1.09E-04	100	100	100	157.62	471.92	132.16	34.72	37.68	32.54
	10	30	1.66E-04	5.42E-04	3.14E-02	100	100	78	337.00	585.44	518.51	6.04	1.32	0.05
$f_3$	2	30	1.67E-10	3.83E-09	1.16E-06	96	94	98	483.00	537.66	528.96	28.29	27.45	23.73
	2	50	3.76E-15	3.55E-12	4.02E-09	100	100	98	552.56	563.84	442.55	48.48	48.16	45.78
$f_4$	5	30	1.26E-06	2.41E-06	1.11E-08	12	72	74	539.50	672.61	438.78	26.17	27.22	28.30
	5	50	9.42E-07	7.36E-09	1.01E-11	28	72	84	449.46	736.11	426.12	44.69	47.78	48.67
	10	30	—	—	4.02E-08	0	0	12	—	—	780.17	—	—	28.83
$f_5$	5	30	1.42E-03	2.96E-04	4.30E-04	62	90	88	277.87	435.66	368.86	19.16	22.11	22.05
	5	50	4.80E-01	1.67E-04	1.30E-03	72	88	98	349.26	443.45	332.51	35.90	38.50	36.55
	10	30	5.46E-03	4.98E-03	5.47E-03	14	30	52	240.42	460.80	437.34	14.71	16.73	16.80

粒子个数设定为 30, 搜索空间为 5 维. 用 3 种方法对 4 个测试函数在一次运算后达到终止条件所需的迭代次数进行了比较. 从图示结果可以看出, 与 PSO-CF 和 PSO-TVIW 方法相比, 使用本文提出的 PSO-L 方法, VFF 达到所设定的阈值, 即粒子群搜索到全局最优值点所需要的迭代次数最少, 达到了算法设计的目的, 进一步表明了理论推导和算法设计的有效性.

对于多个比较对象, 利用单因素方差分析 LSD 方法多次对每一对均值检验之后, 拒绝实际上是真实的零假设的可能性会增加. 因此, 为了比较上述实验中所采取的 3 种方法性能的优劣, 本文采取了方差分析中的 Bonferroni 多重比较检验方法. 将 ITL, AVG, PCL 以及 Avg\_fitness (算法迭代结束所得到解的平均值) 作为响应变量, 对 3 种方法进行了多重比较.

对于各响应变量, 3 种方法的性能优劣排序结果见表 3. 括号表示该试验方法在 Bonferroni 多重比较中没有显著性差异 ( $P > 0.05$ ). 从表 3 可以看出:

1) 算法迭代次数 ITL: ① PSO-CF 与 PSO-TVIW 之间除对 Rosenbrock 函数没有显著性差异外, 对于其余函数二者之间都有显著性差异. ② 粒子个数对 3 种

方法的迭代次数没有显著性影响; 搜索空间的维数对 3 种方法的迭代次数有显著性影响, 尤其对 PSO-CF 和 PSO-L 的影响更为显著. ③ PSO-L 与 PSO-CF 和 PSO-TVIW 之间是否有显著性差异, 依赖于所研究的函数和搜索空间的维数. 在大部分情况下, PSO-L 达到收敛条件所经历的迭代次数明显少于 PSO-CF 和 PSO-TVIW.

表 3 3 种算法 4 个测试指标下性能排序

函数	维数	粒子数	ITL	AVG	PCL	Avg_fitness
$f_1$	5	30	312	(21)3	213	(123)
	5	50	312	(21)3	213	(21)3
	10	30	(13)2	(21)3	(32)1	(21)3
$f_2$	5	30	(12)3	(21)3	(23)1	(21)3
	5	50	312	(21)3	(21)3	(21)3
	10	30	132	(12)3	1(23)	(12)3
$f_3$	2	30	(123)	(123)	(12)3	(12)3
	2	50	3(12)	(123)	(12)3	(12)3
$f_4$	5	30	(31)2	3(21)	3(21)	3(21)
	5	50	(31)2	(32)1	(32)1	3(21)
$f_5$	5	30	132	(23)1	(23)1	(123)
	5	50	(31)2	2(31)	2(31)	(321)
	10	30	2(31)	(32)1	(32)1	(123)

2) 全局最优值 AVG: ① 对于单极值函数, 3 种方法所得到的全局最优值有显著性差异, PSO-L 的性能

劣于 PSO-CF 和 PSO-TVIW. ② 对于多极值函数, 3 种方法所得到的全局最优值没有显著性差异.

3) 收敛到全局最优点的粒子个数 PCL: ① 粒子个数对 3 种方法中收敛到全局最优点的粒子占全体粒子数的比例没有显著性影响; 搜索空间的维数对 3 种方法收敛到全局最优点的粒子个数有显著性影响. ② 对于大部分单极值函数, 迭代结束后, PSO-L 与 PSO-CF 和 PSO-TVIW 有显著性差异, PSO-L 中收敛到全局最优的粒子个数少于其他两种方法. ③ 对于多极值函数, 迭代结束后与其他两种方法比较, PSO-L 中收敛到全局最优的粒子个数最多或处于最优集合中.

4) 算法迭代结束后所得到的解的平均值 Avg fitness: ① 对于所有函数, PSO-CF 与 PSO-TVIW 之间均无显著性差异, PSO-L 与其他 2 种方法间有显著性差异. ② 对于单极值函数, 迭代结束后, PSO-L 得到的解的平均值最大, 其性能劣于 PSO-CF 和 PSO-TVIW. ③ 对于多极值函数, 迭代结束后, PSO-L 得到的解的平均值最小或处于最优集合中.

综上所述, 可得到如下结论:

1) 在处理单极值函数时, 本文提出的方法可以经过更少的迭代次数达到与 PSO-CF 和 PSO-TVIW 同样的收敛率. 但是, 由于迭代次数的大幅减少, 使得其他 3 个相应变量值(全局最优值、达到全局最优粒子个数以及解的均值)不如其他 2 种方法.

2) 在处理多极值函数时, 本文提出的方法可以经过更少的迭代次数达到比 PSO-CF 和 PSO-TVIW 各个响应变量更优的解.

## 5 结 论

本文在以往文献得到的单个粒子轨迹稳定收敛参数取值条件基础上, 通过借鉴和模仿自然界中的个体领导作用, 提出了一种 PSO 改进算法. 将单个粒子的行为和群体行为有机结合, 完成了一个对 PSO 算法从微观层面到宏观层面的改进过程. 通过使用标准函数测试, 从与 2 种经典算法的比较结果看, 虽然新算法的收敛精度等指标在单峰函数上不如经典算法, 但新算法具有收敛率高、迭代次数少等特点, 能够有效减少迭代时间, 增强全局搜索能力. 限于篇幅, 本文算法在模拟电路故障诊断以及结合神经网络在石油探井技术等具体工作中的应用将另文讨论.

## 参考文献(References)

[1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the 6th Int Symposium on Micro Machine and Human Science. Nagoya, 1995: 39-43.  
[2] Shi Y, Eberhart R. A modified particle swarm optimizer[C]. Proc of the IEEE Congress on Evolutionary

Computation(CEC 1998). Piscataway, 1998: 69-73.  
[3] Parsopoulos K E, Vrahatis M N. On the computation of all global minimizers through particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 211-224.  
[4] Kennedy J. The particle swarm: Social adaptation of knowledge[C]. Proc of 1997 Int Conf on Evolutionary Computation. Indianapolis, 1997: 303-308.  
[5] Ozcan E, Mohan C K. Particle swarm optimization: Surfing the waves[C]. Proc of the 1999 Congress on Evolutionary Computation. Washington, 1999, 3: 1939-1944.  
[6] 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.  
[7] 李宁, 孙德宝, 邹彤, 等. 基于差分方程的 PSO 算法粒子运动轨迹分析[J]. 计算机学报, 2006, 29(11): 2052-2061. (Li N, Sun D B, Zou T, et al. An analysis for a particle's trajectory of PSO based on difference equation[J]. Chinese J of Computers, 2006, 29(11): 2052-2061.)  
[8] Samal N R, Konar A, Das S, et al. A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence[C]. IEEE Congress on Evolutionary Computation. Singapore, 2007: 1769-1776.  
[9] Samal N R, Konar A, Nagar A. Stability analysis and parameter selection of a particle swarm optimizer in a dynamic environment[C]. Second UKSIM European Symposium on Computer Modeling and Simulation. Liverpool, 2008: 21-27.  
[10] Xie X F, Zhang W J, Yang Z L. A dissipative particle swarm optimization[C]. Proc of the IEEE Int'l Conf on Evolutionary Computation. Honolulu, 2002: 1456-1461.  
[11] Asanga Ratnaweera, Saman K Halgamuge, Harry C Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.  
[12] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420. (Lv Z S, Hou Z R. Particle swarm optimization with adaptive mutation[J]. Acta Electronica Sinica, 2004, 32(3): 416-420.)  
[13] 吕艳萍, 李绍滋, 陈水利, 等. 自适应扩散混合变异机制微粒群算法[J]. 软件学报, 2007, 18(11): 2740-2751. (Lv Y P, Li S Z, Chen S L, et al. Particle swarm optimization based on adaptive diffusion and hybrid mutation[J]. J of Software, 2007, 18(11): 2740-2751.)