

# 智能单粒子优化算法

纪 震<sup>1)</sup> 周家锐<sup>1)</sup> 廖惠连<sup>2)</sup> 吴青华<sup>2)</sup>

<sup>1)</sup>(深圳大学计算机与软件学院德州仪器 DSPs 实验室 深圳 518060)

<sup>2)</sup>(利物浦大学电气电子工程系 利物浦 L69 3GJ 英国)

**摘 要** 文中在传统粒子群优化(Particle Swarm Optimization, PSO)算法的基础上,提出了智能单粒子优化算法(Intelligent Single Particle Optimizer, ISPO). 与传统的 PSO 算法不同,该算法采用了一个粒子在解空间中搜索,粒子的位置矢量被分成一定数量的子矢量,并基于子矢量对粒子进行更新. 在子矢量更新过程中,通过分析之前的速度更新情况,引入一种新的学习策略,使粒子在搜索空间中能够动态地调整速度和位置,从而向全局最优靠近. 实验表明,此算法对大部分标准复合测试函数都具有很强的全局搜索能力,其寻优能力超过了国际上最近提出的基于 PSO 的改进算法.

**关键词** 智能单粒子优化算法;粒子群优化;子矢量;学习策略

**中图法分类号** TP18

**DOI 号:** 10.3724/SP.J.1016.2010.00556

## A Novel Intelligent Single Particle Optimizer

JI Zhen<sup>1)</sup> ZHOU Jia-Rui<sup>1)</sup> LIAO Hui-Lian<sup>2)</sup> WU Qing-Hua<sup>2)</sup>

<sup>1)</sup>(Texas Instruments DSPs Laboratory, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060)

<sup>2)</sup>(Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, L69 3GJ, UK)

**Abstract** Intelligent single particle optimizer (ISPO) is proposed based on conventional particle swarm optimization (PSO). ISPO applies a particle, which is different from conventional PSO, to search in the problem space. The whole position vector of particle is split into a certain number of subvectors, and the particle is updated based on these subvectors. During the process of updating each subvector, a novel learning strategy is introduced based on the analysis of previous velocity subvectors, and the particle adjusts its velocity and position subvector dynamically. Experimental results demonstrate that ISPO has an outstanding ability to find the global optimum. ISPO performs much better than most recently proposed PSO-based algorithms on the optimization of most complicated composition test functions.

**Keywords** intelligent single particle optimizer; particle swarm optimization; subvector; learning strategy

## 1 引 言

1995 年 Eberhart 博士和 Kennedy<sup>[1]</sup> 博士基于鸟

群觅食行为提出了粒子群优化算法(Particle Swarm Optimization, PSO). 由于该算法概念简明、实现方便、收敛速度快、参数设置少,是一种高效的搜索算法,近年来受到学术界的广泛重视. 但在优化复杂函

收稿日期:2007-09-18;最终修改稿收到日期:2009-06-03. 本课题得到国家自然科学基金(60572100,60872125)、国家自然科学基金委员会与英国皇家学会合作研究项目(60711130233)和深圳市科技三项经费(200704)资助. 纪 震,男,1973 年生,博士,教授,博士生导师,主要研究兴趣为数字信号处理、智能计算、嵌入式系统. E-mail: jizhen@szu.edu.cn. 周家锐,男,1984 年生,硕士研究生,主要研究兴趣为生物启发式算法、DNA 数据压缩技术. 廖惠连,女,1983 年生,博士研究生,主要研究兴趣为智能计算及其应用. 吴青华,男,1953 年生,博士,教授,博士生导师,主要研究兴趣为自适应控制、智能计算、电力系统控制.

数时, PSO 算法很容易陷入局部最优, 并出现早熟收敛现象. 为了提高算法的收敛性能, 随后出现了各种基于不同思想的改进算法. 其中在对于基本粒子群算法进化方程的改进方面, Shi 和 Eberhart<sup>[2]</sup> 于 1998 年对 PSO 算法的速度项引入了惯性权重, 并提出在进化过程中动态调整惯性权重以平衡收敛的全局性和收敛速度. 后来 Shi 和 Eberhart<sup>[3]</sup> 又提出了基于模糊系统的惯性权重的动态调整, 从而实现了对惯性权重的非线性控制. 此外 Clerc 等<sup>[4-5]</sup> 提出带收缩因子的粒子群算法, 以确保算法收敛. 在借鉴其它优化方法的思想方面, 也出现了各类的改进粒子群算法, 如 Angeline<sup>[6]</sup> 于 1999 年借鉴进化计算中的选择概念, 将其引入到 PSO 算法中, 从而淘汰差的粒子, 将具有较高适应值的粒子进行复制以产生等数额的粒子来提高算法的收敛性, 随后 Angeline<sup>[7]</sup> 又提出了借鉴杂交的改进粒子群算法. 在利用小生境思想方面, Suganthan<sup>[8]</sup> 于 1999 年提出一种基于领域思想的粒子群算法, 其基本思想是在算法开始阶段, 每个个体的领域为其自身, 随着进化代数的增长, 其领域范围也在不断增大直至整个种群. 为进一步改善算法性能, 避免过早收敛的现象, Kennedy<sup>[9]</sup> 于 1999 年提出几种基本的领域结构, 如环形结构和轮形结构及它们的推广. 近来性能较为显著的基于 PSO 算法的改进算法有 Peram<sup>[10]</sup> 等人于 2003 年提出的基于粒子群优化的适应值-距离-比例算法 (Fitness-Distance-Ratio based Particle Swarm Optimization, FDR-PSO), 在此算法中每个粒子根据一定的适应值-距离-比例原则, 向附近具有较好适应值的多个粒子进行不同程度的靠近, 而不仅仅只向当前所发现的最好粒子靠近. 此算法改善了 PSO 算法的早熟收敛问题, 在优化复杂函数方面, 其性能得到了较大改善. Bergh<sup>[11]</sup> 等人于 2004 年提出了协同粒子群优化算法 (Cooperative Particle Swarm Optimizer, CPSO), 采用多个协同工作的子粒子群对解向量的不同部分分别进行优化, 达到较好的寻优结果. 为防止 PSO 算法陷入局部最优, Liang<sup>[12]</sup> 等人于 2006 年提出了综合学习粒子群优化算法 (Comprehensive Learning Particle Swarm Optimizer, CLPSO), 使得每个粒子的速度更新基于所有其它粒子的历史最优位置, 从而达到综合学习的目的. 但上述算法在优化复杂高维多模函数时, 容易陷入局部最优, 且解与全局最优值相差较大.

本文基于传统粒子群算法, 提出了智能单粒子优化算法 (Intelligent Single Particle Optimizer,

ISPO). 在 ISPO 算法的优化过程中, 算法不是对整个速度矢量或位置矢量同时进行更新, 而是先把整个矢量分成若干子矢量, 并按顺序循环更新每个子矢量. 在子矢量的更新过程中, 此算法通过引入一种新的学习策略, 使得粒子在更新过程中能够分析之前的速度更新情况, 并决定子矢量在下一次迭代中的速度. 而在传统粒子群算法中, 粒子只是简单的个体, 不具备分析之前速度更新情况的能力. 实验结果表明, 此算法在优化复杂的具有大量局部最优点的高维多模函数方面具有一定的优势, 其性能显著优于最近提出的粒子群改进算法的性能, 且其解非常接近全局最优点.

## 2 PSO 算法

在一个  $D$  维的目标搜索空间中, 由  $n$  个粒子构成一个群体, 其中第  $i$  个粒子 ( $i=1, 2, \dots, n$ ) 的位置可表示为  $D$  维的位置矢量  $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{id}, \dots, z_{iD})$ .  $n$  也被称为群体规模, 过大的  $n$  会影响算法的运算速度和收敛性. 根据一定标准计算  $z_i$  当前的适应值, 即可衡量粒子位置的优劣. 每次迭代中粒子  $i$  移动的距离为粒子的飞行速度或者速度矢量, 表示为  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{id}, \dots, v_{iD})$ , 粒子迄今为止搜索到的最优位置为  $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{id}, \dots, p_{iD})$ , 整个粒子群迄今为止搜索到的最优位置为  $\mathbf{p}_g = (p_{g1}, p_{g2}, \dots, p_{gd}, \dots, p_{gD})$ . 每次迭代中, 粒子根据以下式子更新速度和位置:

$$v_{id}^{k'+1} = \omega v_{id}^{k'} + c_1 \mathbf{r}_1 (p_{id} - z_{id}^{k'}) + c_2 \mathbf{r}_2 (p_{gd} - z_{id}^{k'}) \quad (1)$$

$$z_{id}^{k'+1} = z_{id}^{k'} + v_{id}^{k'+1} \quad (2)$$

其中  $i=1, 2, \dots, n$ ,  $k'$  是迭代次数,  $\mathbf{r}_1$  和  $\mathbf{r}_2$  为  $[0, 1]$  之间的随机数构成的矢量.  $c_1$  和  $c_2$  为学习因子, 也称加速因子, 其使粒子具有自我总结和向群体中优秀个体学习的能力, 从而向自己的历史最优点以及群体内历史最优点靠近. 速度  $v_{id}$  的取值范围为  $[v_{\min}, v_{\max}]$ , 位置  $z_{id}$  的取值范围为  $[z_{\min}, z_{\max}]$ .

## 3 智能单粒子优化算法

在 ISPO 算法的优化过程中, 算法并不是对整个速度矢量或位置矢量 (即所有维的数值) 同时进行更新, 而是先把整个矢量分成若干个子矢量, 并基于子矢量进行粒子更新. 在子矢量的更新过程中, 通过引入一种新的学习策略, 使其对之前的速度更新情

况进行分析,并决定下一次迭代的速度,从而实现速度的动态调整.

3.1 子矢量

大部分随机优化算法(包括粒子群算法和遗传算法等)的性能随维数增加而变差.在更新过程中,传统的 PSO 算法往往同时改变整个解矢量中各维的数值,并根据更新后的解矢量得到一个适应值(fitness value),从而判断解矢量的适应程度.此适应值能够判断解矢量的整体质量,但不能判断部分维是否向最优方向移动.例如,在解决三维函数问题中,其全局最优为 $[0,0,0]$ ,解的初始值设为 $[1,1,1]$ ,给其一个随机扰动 $[0.2,-0.5,0.3]$ ,可得到更新后的解为 $[1.2,0.5,1.3]$ .假设更新后的解的适应值比初始值对应的适应值有所提高,则在下次迭代中解将会在某种程度上向 $[0.2,-0.5,0.3]$ 方向移动.此时,虽然第二维的数值向全局最优靠近,但第一维和第三维的数值却远离了全局最优.因此,对于高维函数,一般 PSO 算法很难兼顾所有维的优化方向.为解决这个问题,在保证粒子能够搜索到空间中的每个区域的同时,可把搜索空间分解成若干个低维小空间进行搜索.

在本文提出的 ISPO 算法中,一个粒子代表着整个位置矢量.在更新过程中,把整个  $D$  维的解空间分成  $m$  部分,即把整个位置矢量分成  $m$  个位置子矢量,其中每一位子矢量与其对应的速度子矢量分别表示为  $\mathbf{z}_j$  和  $\mathbf{v}_j, j=1,\cdots,m$ .为简单起见,设  $D$  刚好被  $m$  整除,则每个子矢量包括了  $l(l=D/m)$  维,如图 1 所示.

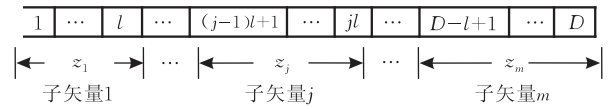


图 1 位置子矢量示意图

对于维之间相关性较大的函数,需根据函数特征来决定子矢量的个数.如图 2 所示,假设全局最优点为 $(0,0)$ ,而 $(k_1,k_2)$ 为直线  $x=k_1$  和  $y=k_2$  上的最优解.把初始值设在 $(k_1,k_2)$ ,并在更新过程中轮流更新每一维.在更新其中一维的数值时需保持另一维的数值不变.由于 $(k_1,k_2)$ 为两直线上的最优解,所以对这两维数值进行轮流更新将会导致解跳不出局部最优 $(k_1,k_2)$ .此时,如果同时改变两维的数值,解有可能会跳出局部最优,并到达全局最优.因此在解决不同问题时,需根据维之间的相关性来决定每个子矢量所包含的维数.

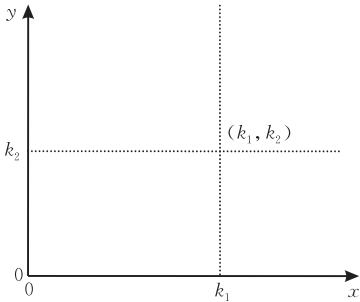


图 2 维相关性示意图

3.2 子矢量的更新过程

ISPO 的更新过程是基于子矢量,按先后顺序(从  $\mathbf{z}_1$  到  $\mathbf{z}_m$ )进行循环更新.在更新第  $j(1\leq j\leq m)$  个子矢量的过程中,将按以下的速度和位置更新公式迭代执行  $N$  次:

$$\mathbf{v}_j^k = (a/k^p) \times \mathbf{r} + b \times \mathbf{L}_j^{k-1} \tag{3}$$

$$\mathbf{z}_j^k = \begin{cases} \mathbf{z}_j^{k-1} + \mathbf{v}_j^k, & f(\mathbf{x}_1^k) > f(\mathbf{x}_2^k) \\ \mathbf{z}_j^{k-1}, & f(\mathbf{x}_1^k) \leq f(\mathbf{x}_2^k) \end{cases} \tag{4}$$

$$\mathbf{L}_j^k = \begin{cases} \mathbf{L}_j^{k-1}/s, & f(\mathbf{x}_1^k) \leq f(\mathbf{x}_2^k) \\ \mathbf{v}_j^k, & f(\mathbf{x}_1^k) > f(\mathbf{x}_2^k) \end{cases} \tag{5}$$

其中  $\mathbf{x}_1^k = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_j^{k-1}, \cdots, \mathbf{z}_m]; \mathbf{x}_2^k = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_j^{k-1} + \mathbf{v}_j^k, \cdots, \mathbf{z}_m]; k=1,2,\cdots,N$ .参数  $\mathbf{L}$  为学习变量,用于分析之前速度的更新情况,从而决定下一次迭代的速度;随机矢量  $\mathbf{r}$  在 $[-0.5,0.5]$ 范围内服从均匀分布;多样性因子  $a$ 、下降因子  $p$ 、收缩因子  $s$  和加速度因子  $b(b\geq 1)$  为常数; $f()$  为评估算法性能的适应值.

在子矢量更新过程中,位置子矢量由速度子矢量决定.速度子矢量由两部分组成:多样性部分  $(a/k^p) \times \mathbf{r}$  和学习部分  $b \times \mathbf{L}_j^k$ .在多样性部分中, $a$  控制随机矢量  $\mathbf{r}$  的幅度,而  $p$  控制幅度的下降梯度.由于 $(a/k^p)$ 是随迭代次数增加而下降的幂函数,所以粒子在优化前期更趋向于全局搜索,并随更新过程逐渐从全局搜索向局部搜索转换.学习部分  $b \times \mathbf{L}_j^k$  完成一种新的学习策略,其中学习变量  $\mathbf{L}$  能够根据之前的速度更新情况而动态地调整速度子矢量.学习部分在更新过程中所起的作用主要为如下 3 点:

(1) 在每次迭代中,如果一个粒子找到比历史最好位置更优的位置时,它将会把这次迭代的速度增加到  $b$  倍,并决定下一次迭代的速度,如式(4)、(5)所示.

(2) 如果在第 1 次迭代中,适应值得到改善,但在第 2 次迭代中适应值没有得到改善.这是因为第 2 次迭代的速度子矢量是第 1 次迭代的速度子矢量

的  $b$  倍,其速度太大以至于位置子矢量跳过了最优点.在这种情况下,如果  $\mathbf{L} \neq \mathbf{0}$ ,降低速度将会有助于搜索全局最优点,因此第 3 次迭代中的学习子矢量  $\mathbf{L}$  将减小为原值的  $1/s$ .

(3) 如果粒子经过几次迭代后适应值仍然没有得到改善,  $\mathbf{L}$  将会被减到一个很小的值,当其小于  $\varepsilon$  时将被设为 0,此时只有多样性部分  $(a/k^p) \times \mathbf{r}$  决定着速度子矢量.这意味着此时的速度会具有更大的多样性,更容易跳出局部最优.

从上述几点可见此粒子的智能性. ISPO 的伪代码如下(其中  $N_{\text{FEs}}$  为函数计算(Function Evaluations, FEs)的次数,  $N_{\text{max}}$  为  $N_{\text{FEs}}$  的最大次数):

开始

初始位置矢量  $\mathbf{z}^0 = [\bar{z}_1^0, \bar{z}_2^0, \dots, \bar{z}_m^0]$ ;

计算适应值  $\mathbf{z}^0$ , 即  $f(\mathbf{z}^0)$ ;

1. 初始 FEs 的次数为  $N_{\text{FEs}} = 1$ ;

2. 初始子矢量的个数  $j = 1$ ;

初始学习变量  $\mathbf{L}_j^0 = \mathbf{0}$ ;

3. 初始子矢量中的迭代次数  $k = 1$ ;

计算  $\bar{\mathbf{v}}_j^k = (a/k^p) \times \mathbf{r} + b \times \mathbf{L}_j^{k-1}$ ;

计算  $f(\mathbf{x}_1^k) = f(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_j^{k-1}, \dots, \bar{z}_m)$ ;

计算  $f(\mathbf{x}_2^k) = f(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_j^{k-1} + \bar{\mathbf{v}}_j^k, \dots, \bar{z}_m)$ ;

$N_{\text{FEs}} = N_{\text{FEs}} + 2$ ;

如果  $f(\mathbf{x}_1^k) > f(\mathbf{x}_2^k)$ , 设置  $\bar{\mathbf{z}}_j^k = \bar{z}_j^{k-1} + \bar{\mathbf{v}}_j^k$ ;  $\mathbf{L}_j^k = \bar{\mathbf{v}}_j^k$ ;

如果  $f(\mathbf{x}_1^k) \leq f(\mathbf{x}_2^k)$ , 设置  $\bar{\mathbf{z}}_j^k = \bar{z}_j^{k-1}$ ;  $\mathbf{L}_j^k = \mathbf{L}_j^{k-1}/s$ ;

如果  $\mathbf{L}_j^k < \varepsilon$ , 设置学习变量  $\mathbf{L}_j^k = \mathbf{0}$ ;

如果  $k < N$ , 跳至步 3;

$j = j + 1$ ;

如果  $j < m$ , 跳至步 2;

如果  $N_{\text{FEs}} < N_{\text{max}}$ , 跳至步 1;

结束.

## 4 实验结果

实验仿真平台为 WindowsXP, Matlab 6.5. 本

实验把 ISPO 算法与带惯性的粒子群算法(Particle Swarm Optimizer with inertia weight  $w$ , PSO- $w$ )<sup>[2]</sup>、协同粒子群优化算法(Cooperative Particle Swarm Optimizer, CPSO)<sup>[11]</sup> 和综合学习粒子群优化算法(Comprehensive Learning Particle Swarm Optimizer, CLPSO)<sup>[12]</sup> 进行比较<sup>①</sup>.

本实验采用了国际上通用的 6 个标准复合测试函数<sup>[13]</sup>:  $CF_1 \sim CF_6$ . 这些测试函数为高维多模函数,具有大量的局部最优点,是优化领域中公认的较难优化的函数.大部分优化算法在对其进行寻优的过程中往往会陷入局部最优点.近年来,优化高维多模函数已成为优化算法领域的研究热点.

在实验中,适应值函数  $f(\cdot)$  的值为当前位置对应的函数值. PSO- $w$  算法的参数设置为  $c_1 = c_2 = 2$ . CPSO 和 CLPSO 的参数设置参考文献[13]. ISPO 算法的参数设置如表 1 所示.

表 1 ISPO 算法的参数设置

函数	$a$	$b$	$p$	$l$	$s$	$\varepsilon$
$CF_1$	4000	2	40	1	4	1.00E+00
$CF_2$	5	2	40	1	4	1.00E+00
$CF_3$	300	2	0.1	1	4	1.00E+00
$CF_4$	4000	2	1	1	4	1.00E+00
$CF_5$	10	2	100	1	4	0.00E+00
$CF_6$	4000	2	1	1	4	1.00E+00

各算法对 6 个标准复合测试函数分别执行 10 次,在每次执行中函数的总计算次数 FEs 设置为  $5 \times 10^4$ ,其平均结果和标准方差如表 2 所示.在优化这 6 个标准复合测试函数时,ISPO 算法都能够到达距全局最优点较近的位置,其算法稳定性较好.与 PSO- $w$ 、CPSO 和 CLPSO 算法相比,ISPO 算法在测试函数  $CF_1$ 、 $CF_2$ 、 $CF_3$  和  $CF_5$  上所得到的最终位置比其它 3 个算法更接近于全局最优点.特别是对于  $CF_1$  和  $CF_5$ , ISPO 算法的适应值明显优于其它 3 个算法. ISPO 的优化结果的标准方差也相对较低,从而证明本算法的稳定性.

表 2 各算法对 6 个标准复合测试函数进行优化的结果

函数	平均结果及标准方差			
	PSO- $w$	CPSO	CLPSO	ISPO
$CF_1$	2.32E+02±1.17E+02	1.39E+02±2.79E+01	1.41E+02±2.13E+02	2.47E+01±7.79E+01
$CF_2$	2.29E+02±1.61E+02	1.15E+02±3.48E+01	1.12E+02±1.40E+02	6.54E+01±6.03E+01
$CF_3$	3.42E+02±1.52E+02	2.27E+02±6.45E+01	3.01E+02±1.95E+02	2.12E+02±6.06E+01
$CF_4$	4.71E+02±1.51E+02	3.77E+02±4.79E+01	3.32E+02±1.26E+02	4.11E+02±5.87E+01
$CF_5$	2.69E+02±2.41E+02	1.11E+02±2.47E+01	1.95E+02±2.36E+02	3.37E+01±2.88E+01
$CF_6$	9.05E+02±3.10E+00	6.71E+02±1.58E+02	8.39E+02±1.44E+02	6.95E+02±2.07E+02

图 3 给出了各算法在优化复合测试函数时的收敛特性,图中横坐标显示 FEs 的范围为  $0 \sim 5 \times 10^4$ ,纵坐标为适应值的对数. ISPO 在  $CF_1$ 、 $CF_2$  和  $CF_5$

上收敛速度明显快于其它 3 个算法.

① 各算法的程序可从 <http://dsp.szu.edu.cn/ps0/ispo> 下载和浏览.

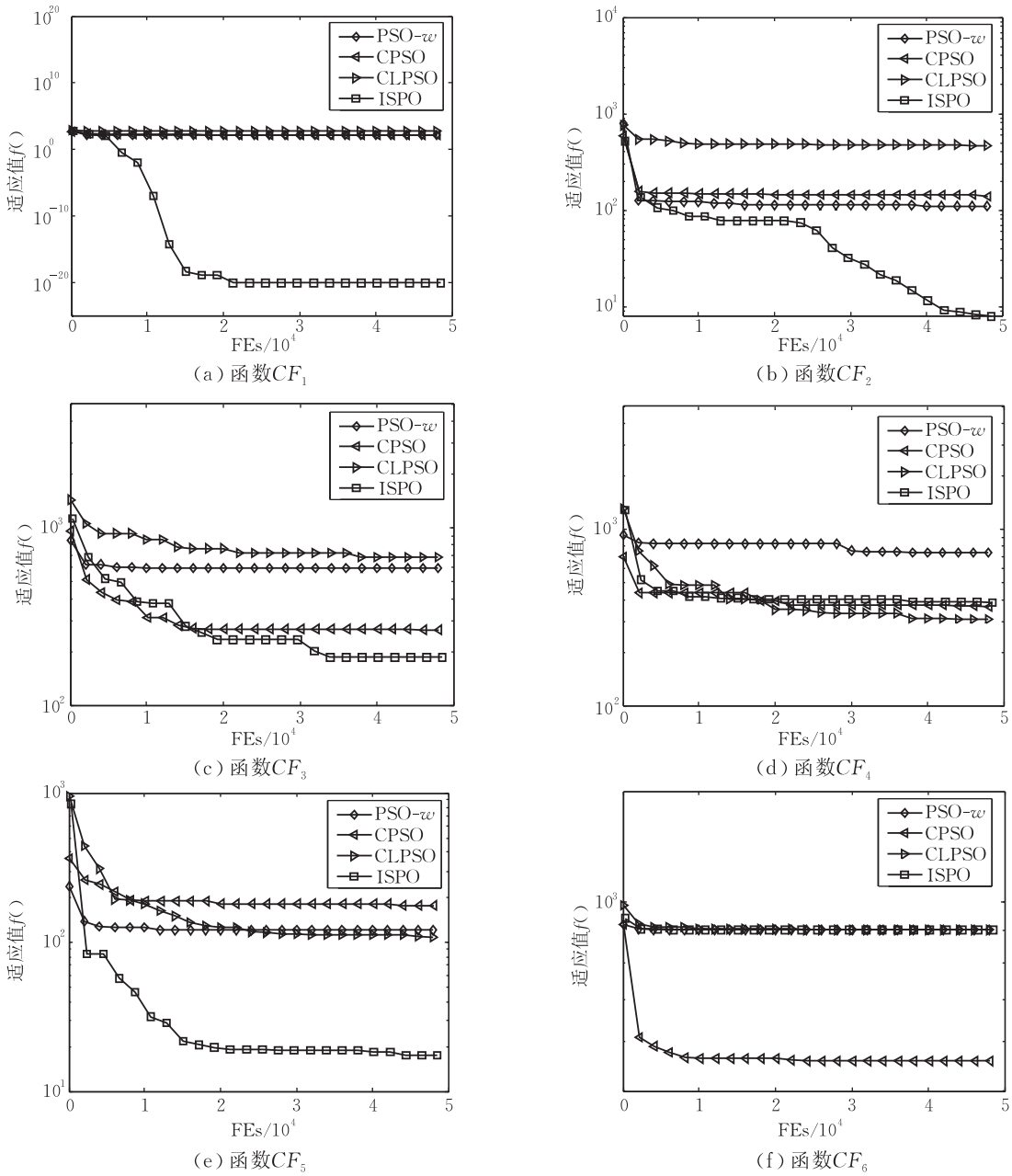


图 3 各算法对标准复合测试函数进行优化的收敛曲线

5 结 论

本文在粒子群优化算法的基础上,针对复杂高维多模函数很难极小化的问题,提出了智能单粒子优化算法,并给出了一种确定算法参数的方法.智能单粒子优化算法先把整个矢量分成若干子矢量,并按顺序循环更新每一子矢量.在更新每一子矢量的过程中,此算法通过引入一种新的学习策略,使粒子能够根据之前的速度更新情况进行动态地调整子矢量速度.粒子在陷入局部最优点时会增加速度的多样性,从而跳出局部最优,并向全局最优靠近.实验

表明,此算法对大部分函数都具有很强的全局搜索能力,特别在优化复杂的高维多模函数时,其性能比起 PSO- $w$ 、FEP 和 CLPSO 算法有着很大的改善,且其解非常接近全局最优点.

参 考 文 献

[1] Kennedy J, Eberhart R C. Particle swarm optimization// Proceedings of the IEEE International Conference on Neural Networks, 1995: 1942-1948

[2] Shi Y, Eberhart R C. A modified particle swarm optimizer// Proceedings of the IEEE International Conference on Evolutionary Computation, 1998: 69-73

- [3] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization//Proceedings of the IEEE Congress on Evolutionary Computation. Seoul, Korea, 2001: 1011-106
- [4] Clerc M. The swarm and the queen: Toward a deterministic and adaptive particle swarm optimization//Proceedings of the Congress on Evolutionary Computation, 1999: 1951-1957
- [5] Corne D, Dorigo M, Glover F. New Ideas in Optimization. McGraw Hill, 1999: 379-387
- [6] Angeline P J. Using selection to improve particle swarm optimization//Proceedings of the IEEE International Conference on Evolutionary Computation. Anchorage, Alaska, USA, 1998: 84-89
- [7] Angeline P J. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences//Proceedings of the 7th Annual Conference on Evolutionary Programming. Germany, 1998: 601-610
- [8] Suganthan P N. Particle swarm optimizer with neighborhood topology on particle swarm performance//Proceedings of the 1999 Congress on Evolutionary Computation, 1999: 1958-1962
- [9] Kennedy J. Small worlds and Mega-minds: Effects of neighborhood topology on particle swarm performance//Proceedings of the Congress on Evolutionary Computation, 1999: 1931-1938
- [10] Peram T, Veeramachaneni K, Mohan C K. Fitness-distance-ratio based particle swarm optimization//Proceedings of the Swarm Intelligence Symposium. Indianapolis, Indiana, USA, 2003: 174-181
- [11] Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 225-239
- [12] Liang J J, Qin A K, Suganthan P N et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295
- [13] Liang J J, Suganthan P N, Deb K. Novel composition test functions for numerical global optimization//Proceedings of the IEEE International Swarm Intelligence Symposium. Messina, Italy, 2005: 68-75



**Ji Zhen**, born in 1973, Ph. D., professor, Ph. D. supervisor. His research interests include digital signal processing, computation intelligence and embedded system design.

**ZHOU Jia-Rui**, born in 1984, M. S. candidate. His research interests include Bio-heuristic algorithm and DNA

compression technique.

**LIAO Hui-Lian**, born in 1983, Ph. D. candidate. Her research interests include computation intelligence and applications.

**WU Qing-Hua**, born in 1953, Ph. D., professor, Ph. D. supervisor. His research interests include adaptive control, mathematical morphology, neural networks, learning systems, evolutionary computation and power system control and operation.

## Background

Particle Swarm Optimization (PSO), proposed by Kennedy and Eberhart in 1995, has been widely used for optimization problems. Since PSO was proposed, research on its improvement and application in engineering optimization becomes a hot spot in the area of evolutionary computation and swarm intelligence. Recently, IEEE Transactions on Evolutionary Computation published a special issue on PSO. PSO has a lot of advantages compared with other evolutionary algorithms, due to its simplicity of implementation, fast convergence rate and easy selection of parameters.

However, for most stochastic optimization algorithms, including PSO, their performance deteriorates as the dimensionality of the search space increases. Many existing PSO-based algorithms are easily trapped into local optima on solving complicated multi-modal functions. In this paper, the new algorithm ISPO, proposed as the improvement of PSO, aims to optimize complicated multimodal functions. ISPO uses a single particle instead of a particle swarm, and applies a novel learning strategy to update the particle's velocity and position. Experimental results have shown that: on solving some widely used multimodal benchmark functions, SPO outperforms PSO with in-

ertia weight, Faster Evolutionary Programming (FEP) and Comprehensive Learning PSO (CLPSO), which was published on IEEE transaction on Evolutionary Computation in 2006.

The research on ISPO is completed by cooperating with a researching group, led by Prof. Zhen Ji in School of Computer Science and Software Engineering, Shenzhen University and Prof. Wu Qinghua in Department of Electrical Engineering and Electronics, The University of Liverpool. The source code of FEP is provided by Yang Z. Y., a student of Yao X. who proposed FEP and was the Editor-in-Chief of the IEEE Transactions on Evolutionary Computation (2003-2008). The source code of CLPSO is provided by Prof. Suganthan, who is the author of the paper CLPSO and is also an associate editor of the IEEE Transactions on Evolutionary Computation.

Another paper entitled "A Strategy of Particle-pair for Vector Quantization in Image Coding", belonging to the same project, has been published on Acta Electronica Sinica, Oct. 2007, 35(10): 1916-1920 by Ji Zhen, Liao Huilian, Xu Wenhuan, Jiang Lai. In this paper, a novel algorithm based on PSO, called particle-pair optimizer (PPO), is proposed to solve vector quantization in image coding.