

一种更简化而高效的粒子群优化算法^{*}

胡 旺⁺, 李志蜀

(四川大学 计算机学院, 四川 成都 610065)

A Simpler and More Effective Particle Swarm Optimization Algorithm

HU Wang⁺, LI Zhi-Shu

(School of Computer Science and Engineering, Sichuan University, Chengdu 610065, China)

+ Corresponding author: Phn: +86-28-85466988, E-mail: scuhuwang@126.com, <http://www.scu.edu.cn>

Hu W, Li ZS. A simpler and more effective particle swarm optimization algorithm. *Journal of Software*, 2007, 18(4):861-868. <http://www.jos.org.cn/1000-9825/18/861.htm>

Abstract: The basic particle swarm optimization (bPSO) has some demerits, such as relapsing into local extremum, slow convergence velocity and low convergence precision in the late evolutionary. Three algorithms, based on the simple evolutionary equations and the extremum disturbed arithmetic operators, are proposed to overcome the demerits of the bPSO. The simple PSO (sPSO) discards the particle velocity and reduces the bPSO from the second order to the first order difference equation. The evolutionary process is only controlled by the variables of the particles position. The extremum disturbed PSO (tPSO) accelerates the particles to overstep the local extremum. The experiment results of some classic benchmark functions show that the sPSO improves extraordinarily the convergence velocity and precision in the evolutionary optimization, and the tPSO can effectively break away from the local extremum. tsPSO, combined the sPSO and tPSO, can obtain the splendid optimization results with smaller population size and evolution generations. The algorithms improve the practicality of the particle swarm optimization.

Key words: evolutionary computation; swarm intelligence; particle swarm optimization; disturbed extremum

摘 要: 针对基本粒子群优化(basic particle swarm optimization,简称 bPSO)算法容易陷入局部极值、进化后期的收敛速度慢和精度低等缺点,采用简化粒子群优化方程和添加极值扰动算子两种策略加以改进,提出了简化粒子群优化(simple particle swarm optimization,简称 sPSO)算法、带极值扰动粒子群优化(extremum disturbed particle swarm optimization,简称 tPSO)算法和基于二者的带极值扰动的简化粒子群优化(extremum disturbed and simple particle swarm optimization,简称 tsPSO)算法。sPSO 去掉了 PSO 进化方程的粒子速度项而使原来的二阶微分方程简化为一阶微分方程,仅由粒子位置控制进化过程,避免了由粒子速度项引起的粒子发散而导致后期收敛变慢和精度低问题。tPSO 增加极值扰动算子可以加快粒子跳出局部极值点而继续优化。对几个经典测试函数进行实验的结果表明,sPSO 能够极大地提高收敛速度和精度;tPSO 能够有效摆脱局部极值点;以上两种策略相结合,tsPSO 以更小的种群数和进化世代数获得了非常好的优化效果,从而使 PSO 算法更加实用化。

关键词: 进化计算;群体智能;粒子群优化;极值扰动

中图法分类号: TP18 文献标识码: A

^{*} Received 2005-11-23; Accepted 2006-04-03

由 Eberhart 和 Kennedy 于 1995 年提出的粒子群优化算法(particle swarm optimization,简称 PSO)^[1]是一种全局优化进化算法,它源于对鸟群和鱼群群体觅食运动行为的模拟.PSO 作为一种并行优化算法,可以用于解决大量非线性、不可微和多峰值的复杂问题优化,并已广泛用于科学和工程领域,如函数优化、神经网络训练、模式分类和模糊系统控制等领域.

与遗传算法类似,粒子群优化算法同样基于群体与适应度这两个概念.粒子群的个体代表问题的一个可能解.每个粒子具有位置与速度两个描述量,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.PSO 算法通过适应度来衡量粒子的优劣.与遗传算法不同,PSO 不是通过遗传算子进化而是通过个体间协作与竞争来寻找最优解.PSO 在进化初期,收敛速度快,运算简单,易于实现,没有遗传算法的编解码和选择、杂交、变异等复杂运算.因此,粒子群优化算法一经提出,立刻引起演化计算领域学者们的广泛关注,并出现了大量的研究成果.

但是,粒子群优化算法根据全体粒子和自身粒子的搜索经验向着最优解的方向发展,在进化后期收敛速度变慢,同时,算法收敛精度不高,尤其是对于高维多极值的复杂优化问题.本文通过简化基本粒子群(basic particle swarm optimization,简称 bPSO)的进化方程和添加极值扰动算子两个改进策略,提出了简化粒子群优化(simple particle swarm optimization,简称 sPSO)算法、带极值扰动的粒子群优化(extremum disturbed particle swarm optimization,简称 tPSO)算法和基于前二者的带极值扰动的简化粒子群优化(extremum disturbed and simple particle swarm optimization,简称 tsPSO)算法.

1 基本粒子群算法及相关改进

bPSO 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:个体极值和全局极值.粒子根据上述两个极值更新自己的速度与位置.在 D 维目标搜索空间中,由种群数为 m 组成粒子群落,其中:第 i 个粒子在第 d 维的位置为 x_{id} ,其飞行速度为 v_{id} ,该粒子当前搜索到的最优位置为 P_{id} ,整个粒子群当前的最优位置为 P_{gd} . Kennedy 和 Eberhart^[1]最早提出的 bPSO 算法公式如下:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (P_{gd} - x_{id}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

其中: $i=1,2,\dots,m;d=1,2,\dots,D;r_1$ 和 r_2 是服从 $U(0,1)$ 分布的随机数;学习因子 c_1 和 c_2 为非负常数,通常取 $c_1=c_2=2$; $v_{id} \in [-v_{\max}, v_{\max}]$, v_{\max} 是由用户设定的常数.迭代终止条件为预设的最大迭代次数或(和)预定的最小适应度阈值.

因为 P_{gd} 为整个粒子群的最优位置,因此,上述算法也被称为全局版 PSO.也可以把第 i 个粒子的部分邻居们搜索到的最优位置作为 P_{gd} ,则相应算法称为局部版 PSO.全局版 PSO 收敛速度快,但有时会陷入局部最优;局部版 PSO 收敛速度相对较慢,但相对不易陷入局部最优值.

Shi^[2]等人通过对式(1)添加动量惯性系数 ω 来提高跳出局部极值能力.修改后的方程如下:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (P_{gd} - x_{id}) \quad (3)$$

Clerc^[3]等人对式(2)添加控制速度的约束因子 α :

$$x_{id}^{t+1} = x_{id}^t + \alpha v_{id}^{t+1} \quad (4)$$

很多学者将式(3)、式(4)视为 bPSO 算法.文献[4]采用模糊规则动态修改 ω 值,使算法自适应地调整全局系数,兼顾了搜索效率和搜索精度.但对许多复杂的非线性优化问题,试图通过自适应调整一个全局系数提高搜索精度的余地是有限的. Angeline^[5]等人借鉴遗传算法思想提出杂交 PSO 算法概念,提高了算法的收敛速度和精度.吕振肃^[6]等人根据粒子群适应度方差作为全局最优化变异条件,提出自适应变异的粒子群优化算法. Van de Bergh^[7]提出了协同 PSO,使粒子更容易跳出局部极小点,达到较高收敛精度,但出现了明显的“启动延迟现象”,在迭代初期减缓了收敛速度. Ratnaweera 等人^[8]在自组织算法的基础上给出了一种变异操作随时间变化的自适应层次 PSO 算法(hierarchical particle swarm optimizer,简称 HPSO),以进一步提高搜索性能,并给出了适合变异操作的自适应参数选择方式.但是, HPSO 算法消除了速度公式中的惯性部分,其发生变异的条件是微粒速度为 0,使微粒不能快速、有效地逃出局部极小点.赫然等人^[9]结合生物界中物种发现生存密度过大会自动分家迁移的习性,给出了一种自适应逃逸微粒群算法,通过逃逸运动,使微粒能够有效地进行全局和局部搜索,减弱了

随机变异操作带来的不稳定性.但是,上述研究都是对基本粒子群方程(1)、方程(2)或方程(3)、方程(4)的改进.

2 简化粒子群优化算法

2.1 关于bPSO中粒子速度项的分析

迄今为止,bPSO 及其改进算法都是基于粒子“位置”和“速度”这两个关键概念,从而在进化方程中都包含位置变量和速度变量,大多数的 PSO 改进算法增加了一些操作算子,如杂交、变异、自适应参数变化等,使得 PSO 算法描述越来越复杂化,同时,也使得定量分析 PSO 的收敛性变得很繁琐^[10-12].

仔细分析 bPSO 的生物模型和进化迭代方程式(3)和方程式(4)可以发现:在 PSO 中,粒子速度概念不是必需的.

从 bPSO 模型角度来看,粒子位置 x_i 代表当前问题的解,优化的最终结果是使 x_i 无限逼近最优解 X^* ,因此,只需要考虑 x_i 的直接变化.粒子速度 v_i 代表粒子移动的快慢程度,粒子移动速度的大小并不代表粒子能够有效趋近最优解位置,反而可能造成粒子偏离正确的进化方向,出现粒子“发散”现象,从而有可能造成后期收敛缓慢,收敛精度低.另外,从式(1)~式(4)来看,位置与速度直接进行加法运算,而没有粒子运动时间概念,这也不符合现实生活中的运动规律 $x=vt$.

定理 1. bPSO 进化过程与粒子速度无关.

证明:文献[3]已经证明,约束粒子速度范围 $v_{id} \in [-v_{max}, v_{max}]$ 与添加约束因子 α 是等价的,因此,可以只考虑由式(3)与式(2)组成的联合进化方程.除 p_{id} 和 p_{gd} 对搜索空间各维的联系以外,每维的更新相互独立.故不失一般性,证明过程可以简化到一维进行,下标 d 可以省略.进一步地,假设种群中除第 i 个粒子外其余粒子保持不动,下标 l 可以省略.再令 $\varphi_1=r_1c_1, \varphi_2=r_2c_2, \varphi=\varphi_1+\varphi_2, \rho = \frac{\varphi_1 P_0 + \varphi_2 P_g}{\varphi_1 + \varphi_2}$.为方便理解,将式(3)和式(2)中变量符的上标移到

变量符后的括号中,则式(3)和式(2)可以变为

$$v(t+1)=\omega v(t)+\varphi(\rho-x(t)) \tag{5}$$

$$x(t+1)=x(t)+v(t+1) \tag{6}$$

将式(5)和式(6)迭代可以得到式(7):

$$x(t+2)+(\varphi-\omega-1)x(t+1)+\omega x(t)=\varphi\rho \tag{7}$$

式(7)是不含速度项的经典二阶微分方程(假设粒子的位置移动为连续过程).

定理 1 的重要性在于说明 bPSO 算法可以没有粒子速度的概念,避免了人为确定参数 $[-v_{max}, v_{max}]$ 而影响粒子的收敛速度和收敛精度.

2.2 简化粒子群优化算法(sPSO)

根据第 2.1 节中的分析,不含速度项的粒子群优化方程可以简化为

$$x_{id}^{t+1} = \omega x_{id}^t + c_1 r_1 (p_{id} - x_{id}^t) + c_2 r_2 (p_{gd} - x_{id}^t) \tag{8}$$

式(8)右边的第 1 项为“历史(history)”部分,表示过去对现在的影响,通过 ω 调节影响程度;第 2 项为“认知(cognition)”部分,表示粒子对自身的思考;第 3 项为“社会(social)”部分,表示与邻居粒子的比较和模仿,实现粒子间的信息共享与合作.

采用第 2.1 节中的符号含义,式(8)可以变形为一阶微分方程:

$$x(t+1)+(\varphi-\omega)x(t)=\varphi\rho \tag{9}$$

由式(7)和式(9)的比较可知:改写后的 sPSO 进化方程不仅没有粒子速度参数,而且方程也由二阶降到了一阶,简化了分析和控制粒子的进化过程.

2.3 sPSO 进化方程的收敛性能分析

定理 2. 若 sPSO 的平均行为按其期望值进行观察,当 $0 < \omega < \frac{c_1 + c_2}{2}$ 时,sPSO 收敛于 $\frac{c_1 P_0 + c_2 P_g}{c_1 + c_2 - 2\omega}$.

证明:式(9)的微分方程解为 $x(t) = Ce^{(\omega-\rho)t} - \frac{\varphi\rho}{\omega-\rho}$, 要使 $\lim_{t \rightarrow \infty} x(t)$ 收敛, 则 $\lim_{t \rightarrow \infty} \left| \frac{x(t+1)}{x(t)} \right| < 1$, 解得 $0 < \omega < \varphi$. 式(8)收

敛于 $\lim_{t \rightarrow \infty} x(t) = \frac{\varphi\rho}{\varphi - \omega}$.

由于 r_1, r_2 服从均匀分布, 故 sPSO 系统的平均行为可通过其期望值进行观察, 即

$$E(c_1 r_1) = c_1 \int_0^1 \frac{x}{1-0} dx = \frac{c_1}{2}, E(c_2 r_2) = c_2 \int_0^1 \frac{x}{1-0} dx = \frac{c_2}{2};$$

故 $\lim_{t \rightarrow \infty} x(t) = p^* = \frac{\varphi\rho}{\varphi - \omega} = \frac{c_1 r_1 p_0 + c_2 r_2 p_g}{c_1 r_1 + c_2 r_2 - \omega} = \frac{c_1 p_0 + c_2 p_g}{c_1 + c_2 - 2\omega}$.

因此, 当 $0 < \omega < \varphi = c_1 r_1 + c_2 r_2 = \frac{c_1 + c_2}{2}$ 时, sPSO 算法收敛于 $\frac{c_1 p_0 + c_2 p_g}{c_1 + c_2 - 2\omega}$.

3 带极值扰动的粒子群优化算法

bPSO 在优化前期中收敛速度很快, 但在优化后期中收敛速度变得缓慢, 因而导致收敛精度低. 这主要是粒子群难以摆脱局部极值的原因, 也是 bPSO 的美中不足. 很多学者提出了许多改进方法, 如杂交 PSO、变异 PSO、自适应 PSO、重新初始化粒子群、结合模拟退火等局部寻优策略等. 但是, 这些策略主要用于调整式(3)、式(4)中的参数 ω, r_1, r_2, α 和变量 x_{id} . 这些改进策略都不同程度地提高了收敛速度和精度, 但都不是从 bPSO 收敛于局部极值的根本原因上采取改进策略. 下面首先分析 bPSO 收敛于局部极值的原因, 然后提出相应的改进策略.

3.1 bPSO 收敛于局部极值的原因分析

实验发现: 当 bPSO 处于进化停滞时, 粒子群中的粒子都会出现“聚集”现象, 直至突破进化停滞局面, 粒子才会“飞散”开去. 下面从理论上解释这一现象.

文献[10]通过严格的数学推导, 得到式(10):

$$\lim_{t \rightarrow +\infty} x(t) = p^* = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2} p_0 + \frac{c_2 r_2}{c_1 r_1 + c_2 r_2} p_g \quad (10)$$

由于 r_1, r_2 服从均匀分布, 故 bPSO 系统的平均行为可通过其期望值进行观察, 故式(10)可以变为

$$\lim_{t \rightarrow +\infty} x(t) = p^* = \frac{c_1 p_0 + c_2 p_g}{c_1 + c_2} = (1 - \alpha) p_0 + \alpha p_g \quad (11)$$

由式(11)和定理 2 可知: 在 bPSO 和 sPSO 中, 粒子将聚集到由自身极值 p_0 和群体全局极值 p_g 决定的极值之上, 如果所有粒子在向 p^* 靠近过程中没有找到优于 p_g 的位置, 则进化过程将处于停滞状态, 粒子逐渐聚集到 p^* .

3.2 带极值扰动的粒子群优化算法

通过第 3.1 节中的分析, bPSO 和 sPSO 中粒子可以通过两种途径摆脱局部极值: (1) 粒子“飞向” p^* 过程中发现比 p_g 更优的解, 但在这一过程中发现更优解的概率较小, 因为 PSO 已经陷入局部极值, p_g 周围已经过高密度的搜索, 难以突破“封锁”, 这也导致了进化后期收敛缓慢; (2) 调整个体极值和全局极值为 p'_0 和 p'_g , 使所有粒子飞向新的位置 p^* , 这使所有粒子从 p^* 迁移聚集到 p^* , 经历新的搜索路径和领域, 因此发现更优解的概率较大.

通过调整参数 ω, r_1, r_2, α 和变量 x_{id} 来优化 bPSO 和 sPSO 后期收敛速度和精度的方法, 属于上述第 1 种情况; 文献[6]的改进方法属于第 2 种情况, 以适应度方差作为触发条件, 并以很小概率 [0.1 0.3] 改变 p_g .

本文采用进化停滞步数 t 作为触发条件, 对个体极值 p_0 和全局极值 p_g 同时进行随机扰动. 极值扰动算子为

$$p_0 = r_3^{t_0 > T_0} p_0; p_g = r_4^{t_g > T_g} p_g,$$

其中: t_0, t_g 分别表示个体极值和全局极值进化停滞步数; T_0, T_g 表示个体极值和全局极值需要扰动的停滞步数阈

值; $r_3^{t_0 > T_0} = \begin{cases} 1, & t_0 \leq T_0 \\ U(0,1), & t_0 > T_0 \end{cases}$ 和 $r_4^{t_g > T_g} = \begin{cases} 1, & t_g \leq T_g \\ U(0,1), & t_g > T_g \end{cases}$ 表示带条件的均匀随机函数.

因此,对 bPSO 方程式(3)和 sPSO 方程式(8)添加极值扰动算子后的形式为

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (r_3^{t_0 > T_0} p_{id} - x_{id}) + c_2 r_2 (r_4^{t_g > T_g} p_{gd} - x_{id}) \quad (3')$$

$$x_{id}^{t+1} = \omega x_{id}^t + c_1 r_1 (r_3^{t_0 > T_0} p_{id} - x_{id}^t) + c_2 r_2 (r_4^{t_g > T_g} p_{gd} - x_{id}^t) \quad (8')$$

本文将式(3')对应的算法称为带极值扰动的粒子群优化算法,简称 tPSO;将式(8')对应的算法称为带极值扰动的简化粒子群优化算法,简称 tsPSO.需要说明的是, T_0 和 T_g 的取值大小决定了极值扰动算子生效的延迟长短,一般取值范围为 3~8.当 T_0 和 T_g 等于进化世代数时,式(3')和式(8')分别退化为式(3)和式(8).因此,tPSO 和 tsPSO 分别是 bPSO 和 sPSO 的推广,bPSO 和 sPSO 分别是 tPSO 和 tsPSO 的特例.

4 实验及结果分析

4.1 实验设计

为了测试本文提出的 sPSO,tPSO 和 tsPSO 的性能,本文设计了 4 类测试实验:(I) bPSO 优化实验;(II) sPSO 优化实验;(III) tPSO 优化实验;(IV) tsPSO 优化实验.实验选用 6 个常用于优化算法比较的基准函数,函数形式、维数、搜索范围、理论极值和优化目标精度见表 1.这些测试函数在不同的文献中有不同的版本,为了方便比较,本文微调了部分函数的形式,使其理论上的最小值都为 0.

Table 1 Benchmark functions used to test the improvement methods

表 1 用于测试改进算法的基准函数

Name and code	Formula	Dim n	Range $[x_{min}, x_{max}]$	Optimal f	Goal for f
Sphere f_1	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^{30}$	0	10^{-5}
Griewank f_2	$f_2(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^{30}$	0	10^{-5}
Rosenbrock f_3	$f_3(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x^2)^2 + (x_i - 1)^2)$	30	$[-100, 100]^{30}$	0	10
Rastrigin f_4	$f_4(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	$[-100, 100]^{30}$	0	10^{-5}
Duadric f_5	$f_5 = \sum_{d=1}^D \left(\sum_{j=1}^D x_j \right)^2$	30	$[-100, 100]^{30}$	0	10^{-5}
Shaffer f_6	$f_6(x) = 0.5 + \frac{(\sin \sqrt{x^2 + y^2}) - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$	2	$[-100, 100]^2$	0	10^{-5}

待优化函数的维数越高、自变量范围越大、目标精度越高,优化过程的难度就越大.为了便于比较和突出改进算法的性能,本文均选用参考文献中最苛刻的函数参数,其具体值见表 1.算法的实验参数也比参考文献中更加严格,具体算法参数设置为:种群数粒子数为 15;进化世代数为 150;最终结果采用独立运行 20 次后的平均值; $c_1=c_2=2; \omega=0.8; T_0=3; T_g=5$.

性能评估采用如下方法:(1) 固定进化迭代次数,评估算法收敛速度和收敛精度;(2) 固定收敛精度目标值,评估算法达到该精度目标所需要的迭代次数;(3) 与文献报道的性能进行比较.

4.2 实验结果及分析

4.2.1 固定进化迭代次数的收敛速度和精度

固定进化迭代次数的实验结果如图 1 所示,其中: a, b, c, d, e, f 分别表示待优化函数 f_1, f_2, f_3, f_4, f_5 和 f_6 的适应度的对数值进化曲线(注:为了方便进化曲线的显示和观察,本文对函数的适应度取以 10 为底的对数;同时,为了避免真数为 0 和纵坐标范围过大,对函数的适应度均加上 10^{-7} 作为截止值).标号 I, II, III, IV 分别表示 bPSO, sPSO, tPSO 和 tsPSO 对以上 6 个函数的进化曲线.

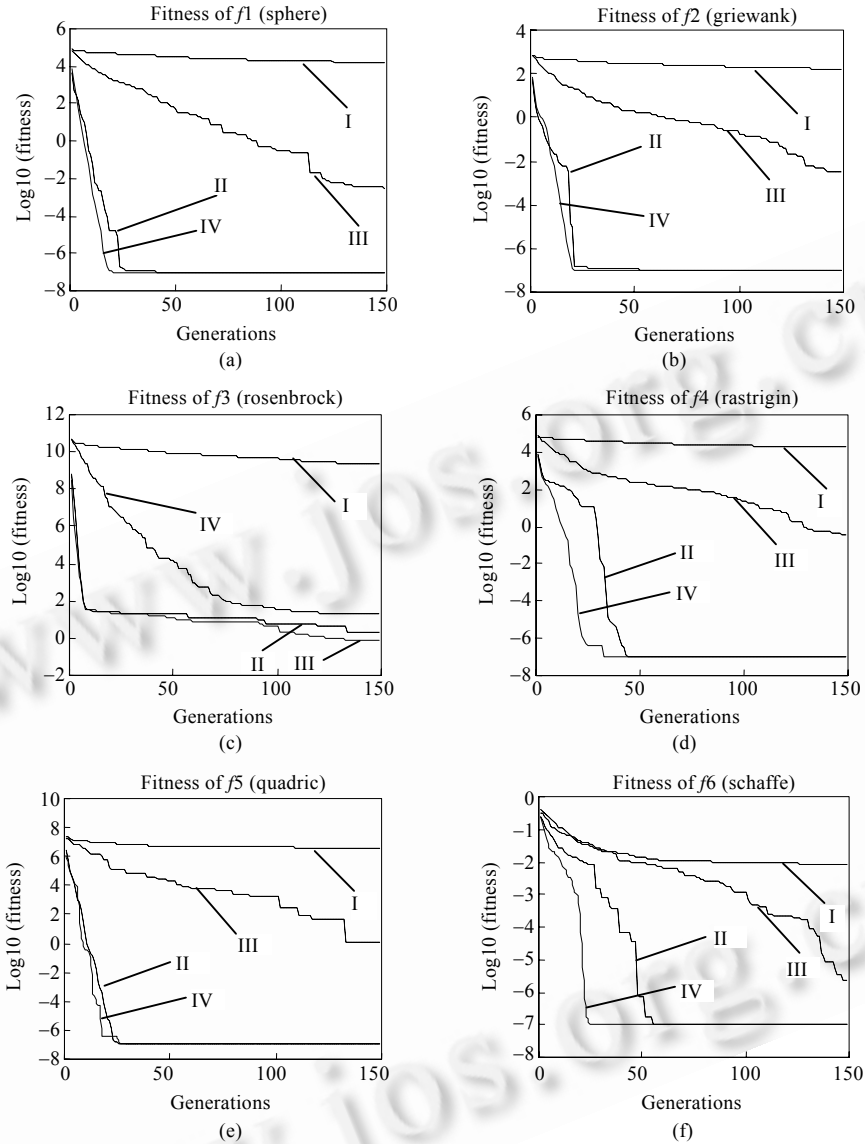


Fig.1 Fitness evolutionary curves of $f_1 \sim f_6$ by the four experiments (I,II,III,IV)

图 1 $f_1 \sim f_6$ 在 4 个实验(I,II,III,IV)中的适应度进化曲线

由图 1 中的曲线(I)和曲线(III)可以看出,sPSO 比 bPSO 在收敛精度和收敛速度上有非常显著的提高,说明了 sPSO 所对应方程(8)的正确性和高效性;由曲线(I)和曲线(III)可以看出,tPSO 比 bPSO 在摆脱局部极值能力、进化后期收敛速度和收敛精度等方面具有显著提高,说明了极值扰动算子具有很好的优化性能;曲线(IV)具有最高的收敛精度和最快的收敛速度,在经过 20~50 次进化迭代后,测试函数基本上达到了目标精度,说明了 tsPSO 算法具有很好的实用性能。

4.2.2 固定收敛精度下的迭代次数

4 个实验在表 1 中指定的收敛精度下经过 20 次独立运行后的平均迭代次数见表 2.其中:成功率(SR)=达到精度的运行次数÷总实验次数;期望迭代次数(ET)=粒子种群数×平均迭代次数÷成功率;“∞”表示期望迭代次数为无穷大;实验代号为“V”是文献[10]报道的结果(注:文献[12]中: f_1, f_2, f_3, f_4, f_6 对应的搜索范围分别为 $\pm 100, \pm 600, \pm 30, \pm 5.12, \pm 100$;它们对应的目标精度分别为 0.01,0.1,100,100, 10^{-5}).

由表 2 中实验代号 I 与 II 可以看出:bPSO 对所有函数都没有达到目标精度;sPSO 除了 f_3 和 f_6 以外均获得 100%的成功率,并且在很少的平均迭代次数(14~24)内就达到了指定的目标精度;sPSO 对于 f_6 也获得了很好的结果,而对 f_3 的优化效果稍差,但还是获得了 20%的成功率.以上结果说明,sPSO 比 bPSO 的优化效果好得多.

同样比较实验代号 I 和 III 可以看出:tPSO 总体上比 bPSO 的优化效果好一些.

实验代号 IV 全部以 100%的成功率达到收敛目标,并且平均迭代次数均在 12~48 次之间,均小于 50 次.这表明,带极值扰动的 sPSO 算法具有非常高的优化性能和实用性.

Table 2 The number of iterations for the goal after 20 runs

表 2 在目标精度下 20 次独立实验的进化代数

Experiment code	Iterations			Success rate	Expected iterations	Experiment code	Iterations			Success rate	Expected iterations		
	Mean	Min	Max				Mean	Min	Max				
f_1	I	150	150	150	0	∞	f_4	I	150	150	150	0	∞
	II	14	12	17	1	210		II	24	17	51	1	360
	III	144	106	150	0.3	7 200		III	149	132	150	0.05	44 700
	IV	13	6	27	1	195		IV	18	9	30	1	270
	V	769	523	1 377	0.4	28 838		V	172	102	308	0.35	7 371
f_2	I	150	150	150	0	∞	f_5	I	150	150	150	0	∞
	II	15	12	19	1	225		II	17	13	19	1	255
	III	147	130	150	0.25	8 820		III	150	150	150	0	∞
	IV	12	6	24	1	180		IV	14	9	18	1	210
	V	689	443	1 589	0.35	29 529		V	-	-	-	-	-
f_3	I	150	150	150	0	∞	f_6	I	150	150	150	0	∞
	II	140	75	150	0.2	10 500		II	84	18	150	0.7	1 800
	III	139	105	150	0.4	5 212		III	110	60	150	0.85	1 941
	IV	48	16	138	1	720		IV	15	7	29	1	225
	V	531	413	695	1	15 120		V	583	63	3 706	0.45	19 433

4.2.3 与参考文献中的优化性能比较

(1) 由表 2 可以看出:除 f_4 的实验 III 外(文献[12]中对 f_4 的搜索范围比本文小了近 40,而目标精度低了 10 倍),实验 II,III,IV 比文献[10]报道的实验代号 V 在函数参数更为苛刻的条件下获得了小得多的期望迭代次数.

(2) 表 3 是本文实验 II,III,IV 与参考文献的实验结果比较.需要特别说明的是,本文实验条件(如粒子种群数、进化迭代次数等)和函数约束条件(如搜索区间、函数维数等)比参考文献中的要苛刻很多.由表 3 可以看出,本文实验 II,III,IV 具有更高的收敛精度.

由上可知,本文提出的 sPSO,tPSO 和 tsPSO 算法比参考文献中的算法具有更好的收敛速度和收敛精度.

Table 3 Comparison of some modified PSOs

表 3 一些改进粒子群算法的性能比较

Result	f_1	f_2	f_3	f_4	f_5	f_6
II	0	0	21.386 96	0	0	0.001 94
III	0.002 57	0.003 19	20.395 62	0.320 59	1.076 75	0.000 002
IV	0	0	0.803 73	0	0	0
Ref.[13]	0	0.002 095	50.798 139	57.1941 36	-	0.000 155
Ref.[5]	9.880 8	0.403 3	1 610.359	46.468 9	-	-
Ref.[10]	0	0.008 614	39.118 488	81.689 550	-	0.002 915
Ref.[14]	-	-	30.316 998	-	-	-

5 结 论

本文从理论上分析和证明了 bPSO 进化方程中的粒子速度概念不是必需的,进而提出了简化版本的 sPSO,使进化方程由二阶降为一阶,并分析了 sPSO 的收敛条件和收敛极值点.同时,本文分析了 bPSO 陷入局部极值的原因,设计了极值扰动算子,使得 bPSO 可以快速摆脱局部极值,并将该极值扰动算子应用于 bPSO 和 sPSO.通过对 6 个经典优化测试函数进行优化实验,结果表明:sPSO 能够极大地提高收敛速度和精度;tPSO 能够有效摆脱局部极值点;以上两种策略相结合,使得 tsPSO 以更小的种群数和进化世代数获得了更好的优化效果,从而使粒子群优化算法更加实用化.

本文的 sPSO 算法已经应用于图像刚性配准中,以两幅待配准图像的规一化互信息作为适应度,获得了很好的应用效果.在相同的种群数和粒子参数下,sPSO 比 bPSO 获得了更高的互信息值(适应度值),表明 sPSO 比 bPSO 具有更高的配准精度;在优化前期,sPSO 比 bPSO 的互信息值上升得更快,表明 sPSO 比 bPSO 具有更快的收敛速度;sPSO 基本上在进化 50~100 代内达到了极大互信息值,而 bPSO 在进化到 150 代时仍未达到 sPSO 的极大值,表明 sPSO 比 bPSO 在优化后期具有更快的配准速度和更高的配准成功率.限于篇幅,本文的算法结合小波分析技术在图像配准中的具体应用将另文发表.

References:

- [1] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proc. of the IEEE Conf. on Neural Networks, IV. Perth: IEEE Press, 1995. 1942-1948. <http://www.engr.iupui.edu/~shi/Coference/psopap4.html>
- [2] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proc. of the IEEE Int'l Conf. of Evolutionary Computation. Piscataway: IEEE Press, 1998. 69-73.
- [3] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: Proc. of the ICEC. Washington, 1999. 1951-1957.
- [4] Shi Y, Eberhart RC. Fuzzy adaptive particle swarm optimization. In: Proc. of the Congress on Evolutionary Computation. Piscataway: IEEE, 2001. 101-106.
- [5] Angeline PJ. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Proc. of the 7th Annual Conf. on Evolutionary Programming. Berlin: Springer-Verlag, 1998. 601-610.
- [6] Lü ZS, Hou ZR. Particle swarm optimization with adaptive mutation. Acta Electronica Sinica, 2004,32(3):416-420 (in Chinese with English abstract).
- [7] van den Bergh F, Engelbrecht AP. Cooperative learning in neural networks using particle swarm optimizers. South African Computer Journal, 2000,(11):84-90.
- [8] Ratnaweera A, Halgamuge SK, Watson HC. Self-Organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans. on Evolutionary Computation, 2004,8(3):240-255.
- [9] He R, Wang YJ, Wang Q, Zhou JH, Hu CY. An improved particle swarm optimization based on self-adaptive escape velocity. Journal of Software, 2005,16(12):2036-2044 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/2036.htm>
- [10] Clerc M, Kennedy J. The particle swarm: Explosion stability and convergence in a multi-dimensional complex space. IEEE Trans. on Evolution Computer, 2002,6(1):58-73.
- [11] Trelea IC. The particle swarm optimization algorithm: Convergence analysis and parameter selection. Information Processing Letters, 2003,85(6):317-325.
- [12] Van den Bergh F. An analysis of particle swarm optimizers [Ph.D. Thesis]. South Africa: University of Pretoria, 2002.
- [13] Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proc. of the 2000 Congress on Evolutionary Computation. Piscataway: IEEE, 2000. 84-88.
- [14] Ke J, Qian JX, Qiao YZ. A modified particle swarm optimization algorithm. Journal of Circuits and System, 2003,8(5):87-91 (in Chinese with English abstract).

附中文参考文献:

- [6] 吕振肃,侯志荣.自适应变异的粒子群优化算法.电子学报,2004,32(3):416-420.
- [9] 赫然,王永吉,王青,周津慧,胡陈勇.一种改进的自适应逃逸微粒群算法及实验分析.软件学报,2005,16(12):2036-2044. <http://www.jos.org.cn/1000-9825/16/2036.htm>
- [14] 柯晶,钱积新,乔谊正.一种改进粒子群优化算法.电路与系统学报,2003,8(5):87-91.



胡旺(1974 -),男,湖南湘潭人,博士生,主要研究领域为计算机应用.



李志蜀(1946 -),男,博士生导师,CCF 高级会员,主要研究领域为计算机网络,多媒体技术,智能控制.