

一种新型多步式位置可选择更新粒子群优化算法

高 芳, 崔 刚, 吴智博, 杨孝宗

(哈尔滨工业大学计算机科学与技术学院, 黑龙江哈尔滨 150001)

摘 要: 粒子群优化(PSO)算法是一种新兴的群体智能优化技术,其由于具有原理简单、参数少、效果好等优点已获得广泛研究和应用.粒子个体极值更新速率低是影响该算法收敛速度和精度的主要因素之一.本文提出一种新型多步式位置可选择更新的粒子群算法,把标准粒子群中速度的单步更新公式分解成三步更新,取所生成的3个位置中的最好位置作为最终结果,细化了粒子的搜索轨迹,在不增加算法复杂度条件下提高了个体极值以及全局极值的更新速率,因而改善了算法的收敛速度和精度.采用 Sphere、Rosenbrock 等6个经典测试函数,并按照固定迭代次数运行和固定时间长度运行两种方法进行测试.测试结果表明该算法简单、稳健、高效,而且明显优于现有的4种经典粒子群算法.

关键词: 智能优化; 粒子群算法; 多步式位置可选择更新策略

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2009) 03-0529-06

A Novel Multi-Step Position-Selectable Updating Particle Swarm Optimization Algorithm

GAO Fang, CUI Gang, WU Zhi-bo, YANG Xiao-zong

(School of Computer Science, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: Particle swarm optimization (PSO) algorithm is a new promising swarm intelligence optimization technology, and it has been extensively studied and applied because of its advantages of simpler theory, less parameters and better performance. However, each particle's individual minimum has a low updating rate, which has been one disadvantageous factor to affect this algorithm speed and precision. In this paper, we propose a novel multi-step position-selectable updating PSO algorithm. This algorithm decomposes the standard PSO velocity single-step updating formula into three steps and selects the best one among the three resultant positions as the final updated position. This scheme refines each particle searching trajectory, increases the updating speed of individual and global minimums, and consequently improves PSO algorithm converging speed and precision without increasing the computing complexity. Six classical testing functions, including Sphere, Rosenbrock and so on, are used to verify the proposed algorithm in two ways: a fixed iteration number test and a fixed time length test. Large numbers of simulations show that the proposed algorithm is simple, robust, and efficient, and meanwhile it outperforms other four existing classical algorithms.

Key words: intelligent optimization; particle swarm algorithm; multi-step position-selectable updating scheme

1 引言

粒子群算法是 Kennedy 和 Eberhart 于 1995 年提出的一种基于群体智能的进化算法^[1].由于其具有原理简单、参数少、易于实现等优点,近年来已成为继遗传算法、蚁群算法之后自然计算领域中的研究热点,并被广泛应用于复杂函数优化、神经网络训练、模式分类、控制系统参数寻优、车间调度等领域.类似于 GA, PSO 算法也存在着精度较低、早熟等缺点,因此许多学者针对 Kennedy 和 Eberhart 的标准粒子群算法(以下简称 PSO-

S)提出了许多改进方法来克服其不足之处,诸如加入微分演化(DE)和交叉操作^[2,3]、引入速度惯性权重^[4,5]、加入局部混沌细化搜索^[6]、以及与 GA、SA 等算法进行融合^[7,8]等,但是都在不同的程度上增加了算法的复杂性.为了提高算法的性能,对算法各项参数的研究越来越复杂,有些研究甚至超过了算法自身的复杂性.笔者认为这样有悖于粒子群算法最初提出的目的:即简单实效性.本文首先简述包括 PSO-S 在内的四种基本粒子群算法,然后从粒子个体极值更新速率的角度分析 PSO-S 存在不足的原因,进一步提出一种新型的多步式位置可

收稿日期:2007-11-20;修回日期:2008-08-12

基金项目:国家“八六三计划”重大课题项目(No. 2006AA01A103)

选择更新的粒子群算法,并通过与四种经典算法仿真对比测试进行验证。

目前,在粒子群的各种改进算法中,通过平衡粒子的“开拓(exploration)”能力与“开掘(exploitation)”能力之间的关系,实现在迭代初期算法具有很强的开拓能力(全局搜索能力)以避免陷入局部极值,在迭代后期算法又能具有很强的开掘能力(局部搜索能力),达到快速收敛。本文在深入分析粒子群算法基本原理、粒子飞行轨迹及个体极值更新速率的基础上,提出一种即简单又高效的改进算法:一种新型粒子矢量位置可选择更新的粒子群优化算法。该算法在不增加算法的复杂性的基础上达到了改进的目的。并通过与上述所提的粒子群四种最优的经典算法相比较,验证本文改进算法有效性和改进思想的先进性。

本文结构如下:首先简述包括 PSO-S 在内的四种基本粒子群算法,然后从粒子个体极值更新速率的角度分析 PSO-S 存在不足的原因,受人类行为的启发,进一步提出一种新型粒子矢量位置可选择更新的粒子群优化算法,最后采用 Rosenbrock 等 6 个经典测试函数,在同一随机初始种群状态下,按照固定迭代次数和固定时间长度两种方法与四种经典粒子群算法进行仿真对比测试,验证本文所提出算法的简单高效性。

2 粒子群算法简介

粒子群算法基本思想源于对鸟群捕食行为的研究,鸟群中每个个体根据自身搜索经验和其它个体的搜索经验两个因素来确定觅食的方向。在 PSO 算法中,粒子通过跟踪两个“极值”来更新自己:第一个是粒子本身所找到的最好解,叫做个体极值点(用 p^{best} 表示),第二个是整个种群找到的最好解,称为全局极值点(用 g^{best} 表示)。那么 PSO-S 的具体迭代公式为

$$v_i^{(k+1)} = v_i^{(k)} + c_1 \cdot r_1 \cdot (pbest_i^{(k)} - x_i^{(k)}) + c_2 \cdot r_2 \cdot (gbest^{(k)} - x_i^{(k)}) \quad (1)$$

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

式中, x_i 表示粒子 i 的当前位置; v_i 表示粒子 i 的当前速度; c_1 、 c_2 为正常数,一般都取 2; r_1 、 r_2 为 $[0, 1]$ 区间的随机数。

所有粒子都由被优化函数 $f(x)$ 评价其适应度值,对于求解 $\min f(x)$ 问题,个体极值点 $pbest_i$ 根据 $f(pbest_i)$ 的大小按下述规则更新:

$$pbest_i^{(k+1)} = \begin{cases} x_i^{(k+1)}, & \text{if } f(x_i^{(k+1)}) < f(pbest_i^{(k)}) \\ pbest_i^{(k)}, & \text{otherwise} \end{cases} \quad (3)$$

全局极值点 $gbest$ 取所有个体极值点中的最好点进行更新。公式(1)右边第一项 v_i 由于具有扩大搜索空

间、探索新的搜索区域的趋势而使算法具有全局优化能力。但是迭代后期 v_i 可能影响局部精细搜索。Shi 和 Eberhart 提出了惯性权重方法^[4],即在式(1)的 v_i 前乘以惯性权重 w ,用参数 w 控制前一次迭代速度在当前速度中所占的比例。Shi 等还进一步提出了惯性权重线性变化的改进算法(简称 PSO-CIV)^[5],即将的初值设置为 0.9,并使其随迭代次数的增加线性递减至 0.4,以使算法实现从迭代初期的全局搜索转向后期的局部精细搜索。另外在该算法中他们还对粒子速度进行钳位,即指定粒子更新的最大速度,当粒子更新速度超过最大速度时,将被钳位在最大速度以防止速度过大而影响搜索精度。最大速度的具体设定如下:

$$v_i^{\max} = r(x^u - x^l) \quad (4)$$

式中, r 为常数,且 $0 < r < 1$; x^u 和 x^l 为粒子搜索空间的上、下界。

Fourie 和 Groenwold 进一步发展了 Shi 和 Eberhart 的方法,提出了动态惯性权重及速度钳位的方法(简称 PSO-DIV)^[9]。动态权重是指如果在一定的迭代代数 h 内全局极值点保持不变而没有被更新,则按一定规则减小惯性权重和最大钳位速度的值,具体规则为

$$\text{if } f(gbest^k) = f(gbest^{k-h}) \text{ then } w_{k+1} = w_k, v_{k+1}^{\max} = v_k^{\max} \quad (5)$$

式中, w 为常数且 $0 < w < 1$ 。

Clerc 和 Kennedy 对 PSO 算法的收敛性进行了分析,并提出采用收敛因子可以确保算法的收敛^[10]。PSO 收敛因子模型(简称 PSO-C)的速度更新公式如下:

$$v_i^{(k+1)} = K [v_i^{(k)} + c_1 \cdot r_1 \cdot (pbest_i^{(k)} - x_i^{(k)}) + c_2 \cdot r_2 \cdot (gbest^{(k)} - x_i^{(k)})] \quad (6)$$

$$\text{式中, } K \text{ 为收敛因子,且 } K = \frac{2}{|2 - c_1 - c_2 - \sqrt{c_1^2 + c_2^2 - 4}|},$$

其中 $c_1 + c_2 > 4$, K 值通常取 4.1,则 $K = 0.73$ 。

Schutte 和 Groenwold 对七种 PSO 改进算法进行了分析^[11],这七种算法中包含了上述 PSO-CIV、PSO-DIV 及 PSO-C 三种方法。他们采用 Dixon Szego 测试集进行仿真测试,测试结果表明 PSO-CIV、PSO-DIV 及 PSO-C 的性能要优于其它四种。本文将在分析粒子群算法个体极值更新速率的基础上提出新型改进算法,并将其与上述三个算法及 PSO-S 作对比测试来检验其性能。

3 个体极值更新速率分析

PSO-S 算法的速度更新公式(1)中包括 $v_i^{(k)}$ 、 $c_1 \cdot r_1 \cdot (pbest_i^{(k)})$ 、 $c_2 \cdot r_2 \cdot (gbest^{(k)} - x_i^{(k)})$ 三项,其中: $pbest_i$ 、 $gbest$ 分别代表粒子的“自身经验”和“社会经验”,粒子通过它们和群体实现协同合作,因此其变化是不可忽视的。已有研究表明 $v_i^{(k)}$ 的方向和长度和 $pbest_i^{(k)}$ 和

$gbest_i^{(k)}$ 之间存在着耦合, 导致 $pbest_i$ 更新缓慢^[2]. 例如求解三维函数 $f(x, y, z) = x^2 + y^2 + z^2$, 其定义域为 $-10 < x, y, z < 10$, 参见图 1. 假设 $x_i^{(k)} = pbest_i^{(k)}$, 也就是说 $pbest_i^{(k)}$ 刚刚用 $x_i^{(k)}$ 更新过, $v_i^{(k+1)}$ 的计算式包括一组完全不同的矢量集: $x_i^{(k-1)}, x_i^{(k)} = pbest_i^{(k)}, pbest_i^{(k-1)}$, 这里取粒子群算法的参数 $c_1 = c_2 = 2$, 则下一步迭代位置为

$$x_i^{(k+2)} = x_i^{(k+1)} + 2r_1(pbest_i^{(k+1)} - x_i^{(k+1)}) + 2r_2(gbest^{(k+1)} - x_i^{(k+1)}) + v_i^{(k+1)} \quad (7)$$

假设 $x_i^{(k+1)} = (x_1, y_1, z_1)$, $pbest_i^{(k+1)} = (x_2, y_2, z_2)$ 分别位于任意两个相邻象限 (如图分别在 II、III 象限), 并且有 $f(x_i^{(k+1)}) > f(pbest_i^{(k+1)})$, $gbest^{(k+1)} = (\frac{-x_2}{1}, -y_2, z_2)$, 其中 $\frac{-x_2}{1}$ 是小的正常数, 根据公式(7)则有

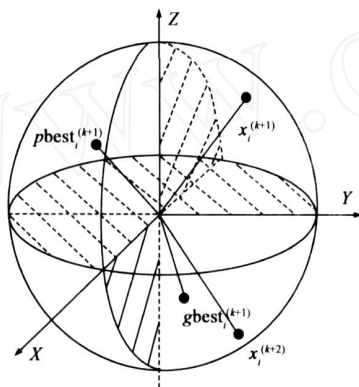


图1 个体极值更新示意图

$$x_i^{(k+2)} = (x_1, y_1, z_1) + v_i^{(k+1)} + \left(-2x_1(r_1 + r_2) + 2x_2 \left(r_1 - \frac{r_2}{1 + \dots} \right), -2y_1(r_1 + r_2) + 2y_2(r_1 - r_2), -2z_1(r_1 + r_2) + 2z_2(r_1 - r_2) \right) \quad (8)$$

由上式不难分析出: 当 $r_1 = 0.6, r_2 = 0.5 + \dots$ 时, $f(x_i^{(k+2)}) > f(pbest_i^{(k+1)})$, 即使 $v_i^{(k+1)}$ 为零, $x_i^{(k+2)}$ 不能改进 $pbest_i^{(k+1)}$, $pbest_i$ 更新缓慢进一步导致 $gbest$ 更新缓慢、搜索效能劣化, 这就是 PSO 收敛慢、精度差的原因。

4 多步式位置可选择更新的粒子群算法

上述分析表明只有提高个体极值的更新率才能提高算法的效率和精度. 针对上述的 v_i^k 和 $pbest_i^k, gbest^k$ 的耦合导致 $pbest$ 更新缓慢的缺点. 在粒子进化过程中, 速度更新公式中的自身开拓能力、自身经验和群体经验三者之间是否密不可分? 受人类社会行为的启发: 我们每一个人在做一件事的过程中, 有时依靠自身经验, 有时依靠群体经验, 有时则是完全的个人创新行为, 而有时则是综合考虑各种经验后再去做. 每一种情况都有得到好结果的可能. 三者是可以分开和任意组合的. 本文在考虑到不增加粒子群算法的复杂性的基础上, 只保存算法中间计算结果和最终计算结果作为可选值进行算法改进: 一种新型多步式位置可选择更新的改进算法 (简称 PSO-MP), 该算法的基本思想是把

标准粒子群中速度的单步更新公式(1)分解成三步更新, 取所生成的 3 个位置中的最好位置作为最终结果. 这种先分步再选择更新的方式可以实现 $v_i^{(k)}, pbest_i^{(k)}$ 和 $gbest^{(k)}$ 解耦, 从而提高算法性能. PSO-MP 算法的具体内容介绍如下: 首先来看 PSO-S 的算法的粒子位置更新过程, 它先是由公式(1)和(2)算出粒子速度和位置, 再计算 $f(x_i^{(k+1)})$, 然后利用公式(3)更新, 其中 $x_i^{(k+1)}$ 是 $x_i^{(k)}$ 与 $v_i^{(k)}, c_1 \cdot r_1 \cdot (pbest_i^{(k)} - x_i^{(k)}), c_2 \cdot r_2 \cdot (gbest^{(k)} - x_i^{(k)})$ 三项的矢量和, 如图 2 所示.

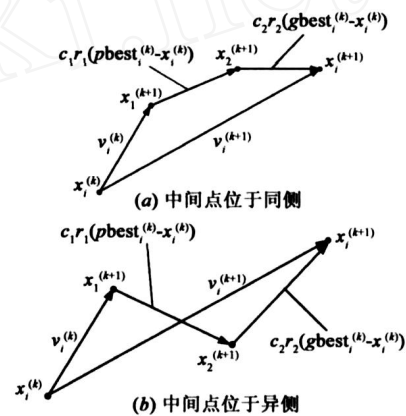


图2 粒子位置更新矢量示意图

由图 2 可知, PSO-S 算法的位置更新只是涉及 $x_i^{(k)}, x_i^{(k+1)}$ 两点, 实际上中间点 $x_1^{(k+1)}, x_2^{(k+1)}$ 在公式(1)的连加式中已被计算过, 这两点有可能优于 $x_i^{(k+1)}$ 但却被作为临时点忽略掉. 因此本文的 PSO-MP 算法将式(1)的单步连加式改为分步计算, 产生以上两个中间点, 它们的表达式分别为

$$x_1^{k+1} = x_i^{(k)} + v_i^{(k+1)} \quad (9)$$

$$x_2^{(k+1)} = x_1^{(k+1)} + c_1 \cdot r_1 \cdot (pbest_i^{(k)} - x_i^{(k)}) \quad (10)$$

接着计算两个中间位置的函数值, 然后把它们和 $f(x_i^{(k+1)})$ 进行比较, 取函数值较小的位置点作为 $x_i^{(k+1)}$ 的更新值. 整个算法的公式描述如下:

$$v_1^{(k+1)} = v_i^{(k)}, x_1^{(k+1)} = x_i^{(k)} + v_1^{(k+1)} \quad (11)$$

$$v_2^{(k+1)} = v_1^{(k+1)} + c_1 \cdot r_1 \cdot (pbest_i^{(k)} - x_i^{(k)}), x_2^{(k+1)} = x_1^{(k+1)} + v_2^{(k+1)} \quad (12)$$

$$v_3^{(k+1)} = v_2^{(k+1)} + c_2 \cdot r_2 \cdot (gbest^{(k)} - x_i^{(k)}), x_3^{(k+1)} = x_2^{(k+1)} + v_3^{(k+1)} \quad (13)$$

$$x_i^{(k+1)} = \begin{cases} x_1^{(k+1)}, & \text{if } f(x_1^{(k+1)}) \\ f(x_2^{(k+1)}) \text{ and } f(x_1^{(k+1)}) < f(x_3^{(k+1)}) \\ x_2^{(k+1)}, & \text{if } f(x_2^{(k+1)}) \\ f(x_1^{(k+1)}) \text{ and } f(x_2^{(k+1)}) < f(x_3^{(k+1)}) \\ x_3^{(k+1)}, & \text{otherwise} \end{cases} \quad (14)$$

$$\text{update}(pbest_i) : \text{same as eq. (3)} \quad (15)$$

另外 PSO-MP 算法在按公式(11)和(12)得出 $v_2^{(k+1)}$ 和 $v_3^{(k+1)}$ 后再采用式(4)进行钳位,然后再更新 $x_2^{(k+1)}$ 和 $x_3^{(k+1)}$.

PSO-MP 算法除了具有前述的解耦作用外,它的有效性进一步体现在:算法首先按照上一步的速度先移动到点 $x_1^{(k+1)}$,此为速度惯性单独作用产生的点,如果上步迭代的搜索方向较好,该点可能是个较好点.然后在点 $x_1^{(k+1)}$ 基础上再向个体极值点运动一步到 $x_2^{(k+1)}$,此步有可能细化局部搜索.再按照趋向于全局极值点的方向运动到 $x_3^{(k+1)}$,该点也是一般粒子群算法确定的更新点 $x_i^{(k+1)}$.最后比较 $x_1^{(k+1)}$, $x_2^{(k+1)}$, $x_3^{(k+1)}$ 并择其中最优者作为第 $k+1$ 次迭代的最终更新点 $x_i^{(k+1)}$.由此可见算法由于具有多重选择机会而更快趋向最优解.

从粒子轨迹角度来分析,PSO-MP 算法通过引入 $x_1^{(k+1)}$ 、 $x_2^{(k+1)}$ 两个中间点相当于对原粒子搜索的点列轨迹进行了加密细化,另外由图 2 可知,原 PSO-S 算法的轨迹为中心迹线,PSO-MP 算法每步在其左右两侧取两点,使原来单一迹线具有一定的搜索“宽度”.通过二维优化问题的粒子轨迹仿真可以发现:粒子的轨迹不断趋向于最优点同时具有随机性,所以粒子经常是直线跨过最优点或以一定偏置距离“绕过”最优点,以上“跨”、“绕”现象在算法后期尤为普遍.而 PSO-MP 算法使粒子轨迹“宽化”,就会使粒子的更新位置更有可能

接近最优点,从而使搜索速度和精度都能提高.

在算法复杂度方面,PSO-MP 算法没有增加循环语句,主要是把粒子速度公式中的三项连加式拆成两个两项加式.算法相对于 PSO-S 所增加的运算量包括:求 $x_1^{(k+1)}$ 和 $x_2^{(k+1)}$ 的两个加式(式(11)和(12)),以及更新 $x_i^{(k+1)}$ 的条件语句(公式(14)),其中包含两次函数调用 $f(x_1^{(k+1)})$ 和 $f(x_2^{(k+1)})$.因此说 PSO-MP 算法的运算量是线性增加,与 PSO-S 算法是同阶的复杂度.

综合上述,PSO-MP 算法承袭了 PSO-S 简单、稳健、运算量小的特点,同时具有更高的搜索效率和精度. PSO-MP 的另一大优点在于其所需调节的参数少,通用性好.前述的惯性权重等算法参数较多,而且还需面向具体问题调整设置参数.而 PSO-MP 将单步位置更新分解为多步更新的过程并未引入参数,只是在采用速度钳位时引入常数 r (参见公式(4),可令 r 恒等于 0.5),参数少且无须调整,因此通用、易操作.

5 仿真测试

本文选取如表 1 所示的 6 个典型函数来测试 PSO-MP 算法的性能,取 PSO-S、PSO-C、PSO-CIV、PSO-DIV 作为对比算法.为了增强可比性,以上五种算法的共同参数取值相同:首先五种算法种群规模都相同,其次 PSO-MP、PSO-S、PSO-CIV、PSO-DIV 的共同参数 $c_1 = 2$ 、 $c_2 = 2$. PSO-C 的 $c_1 = 2.8$ 、 $c_2 = 1.3$. PSO-CIV、PSO-DIV 的参数 $w = 0.6$ 、 $r = 0.5$ 、 $p = 0.99$ 、 $h = 10$.

表 1 测试函数

序号	函数名称	函数表达式 $f(x)$	自变量范围	维数	最小值	最优点
1	Sphere	$\sum_{j=1}^n x_j^2$	$[-100, 100]$	30	0	$(0, 0, \dots, 0)$
2	Rosenbrock	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + x(x_i - 1)^2]$	$[-30, 30]$	10	0	$(1, 1, \dots, 1)$
3	Rastrigin	$\sum_{j=1}^n (x_j^2 - 10\cos(2\pi x_j) + 10)$	$[-2, 2]$	10	0	$(0, 0, \dots, 0)$
4	Griewank	$1 + \frac{1}{400} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]$	10	0	$(0, 0, \dots, 0)$
5	Ackley	$-20 \exp\left(-0.02 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-30, 30]$	10	0	$(0, 0, \dots, 0)$
6	Salomon	$1 - \cos\left(2 \sqrt{\frac{n}{N}} \sum_{j=1}^N x_j^2\right) + 0.1 \sqrt{\frac{n}{N}} \sum_{j=1}^N x_j^2$	$[-100, 100]$	10	0	$(0, 0, \dots, 0)$

在 Matlab7.0 环境中编写 m 文件形式的测试程序,首先以 PSO-S 为对照来测试本文 PSO-MP 算法的个体极值更新率,取最大迭代次数 $n = 5000$,种群大小 S_{size} 取测试函数维数的 10 倍.对每个测试函数,运行以上两个算法各 10 次 ($N_r = 10$),设单次运行某算法个体极值的更新总次数为 N_i ,则该算法的个体极值平均更新率公式为 $Update_rate = \frac{N_i}{\sum_{i=1}^N (N_i / (S_{size} \times n))} / N_r$. 测试结果参

见图 3,由该图可知 PSO-MP 的个体极值更新率基本都达到了 PSO-S 的 2 倍以上.

采用前述参数值对表 1 中的六个测试函数进行测试,采用同一随机初始种群迭代 5000 次,5 种算法的运行过程如图 4 所示.

进一步的仿真测试包括两种类型:第一种是在相同的固定迭代次数下运行,第二种是在相同的固定时间长度内运行,比较上述两种情况下五种算法搜索出

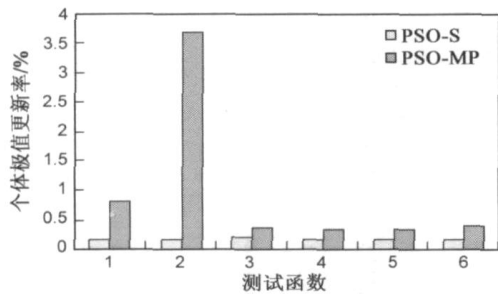


图3 个体极值更新率对比

的最优解的优劣. 以上两种仿真的参数设置参见表 2 和表 3. 为了能够真实比较各种算法的效果, 所有算法每次运行都是采用同一随机初始种群, 并且每个测试函数运行多次, 然后取统计结果进行比较. 具体种群大小、运行次数及时间等参数的设置如表 2 所示.

表 2 仿真的参数设置

函数序号	固定迭代次数运行			固定时间长度运行		
	种群大小	运行次数	迭代次数	种群大小	运行次数	固定时间长度 (s)
1	300	50	1000	300	10	10
2	1000	20	100	100	10	10
3	100	20	500	100	10	10
4	100	50	1000	100	10	10
5	100	20	1000	100	10	10
6	100	100	100	100	10	5

在 HP Pavilion dv2210 笔记本 (Dual Core 2 2.0GHz CPU, 1022 MB RAM) 上按表 2 参数运行测试程序, 记录每次运行结果, 统计出每个算法对于表 1 中 6 个测试函数所求出最优解的均值、标准差和最小值 (即最好值), 结果参见表 3 和表 4.

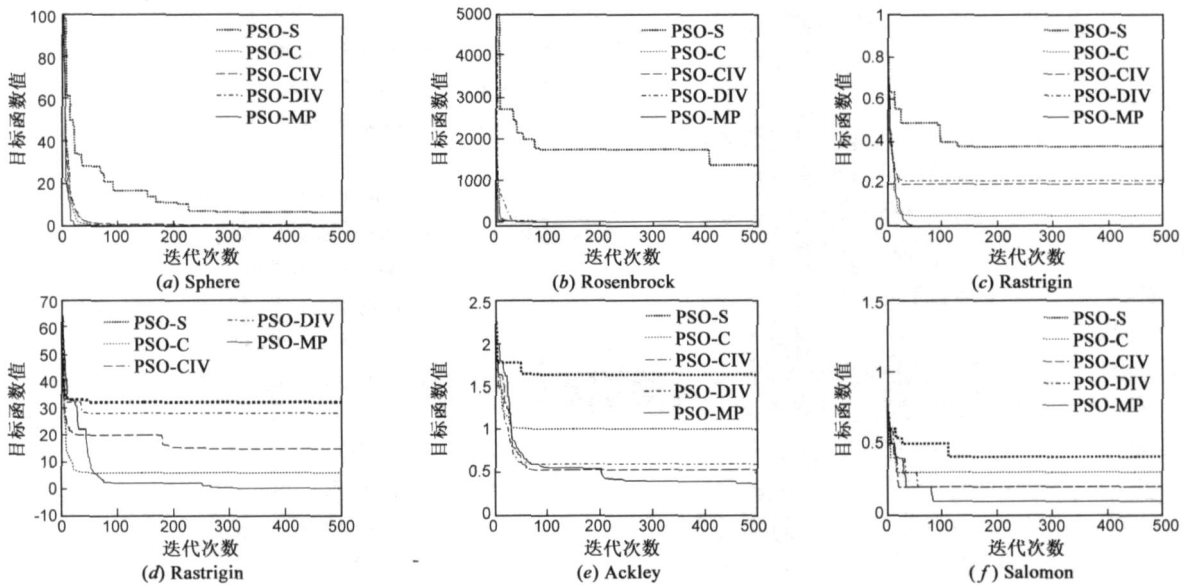


图4 五种算法的Rastrigin函数的测试

表 3 固定迭代次数仿真的结果

算法	测试函数 1			测试函数 2			测试函数 3			测试函数 4			测试函数 5			测试函数 6		
	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值
PSO-S	1076.97	403.24	432.25	1213.23	1493.62	259.95	25.20	5.20	17.03	2.11	0.79	1.09	2.11	0.26	1.58	2.97	0.58	1.60
PSO-C	3.13	3.07	0.29	21.52	34.62	2.69	9.70	5.24	3.98	0.23	0.15	0.020	0.95	0.29	0.41	0.45	0.16	0.20
PSO-CIV	26.95	14.85	4.07	8.61	19.63	0.27	6.02	2.47	1.99	0.48	0.77	0.017	0.71	0.20	0.40	0.43	0.16	0.20
PSO-DIV	29.28	15.77	3.73	8.37	16.48	0.40	5.77	3.34	1.99	0.61	1.26	0.025	0.77	0.23	0.43	0.43	0.16	0.20
PSO-MP	1.09 $\times 10^{-9}$	2.66 $\times 10^{-9}$	0	4.00	1.54	5.55 $\times 10^{-4}$	4.59	4.85	1.11 $\times 10^{-11}$	0.22	0.12	0.016	0.41	0.26	6.40 $\times 10^{-6}$	0.26	0.095	0

由表 3 和 4 可以看出, PSO-MP 算法为以上六个测试函数找到的最优解均值是 5 种算法中最好的, 其中 PSO-MP 除对于测试函数 4 和其它算法较接近外, 而对于其它测试函数 PSO-MP 算法明显好于其它算法, 对于所能达到的最小值, PSO-MP 算法要远远优于其它算法. 这是因为本文的改进算法主要是从提高了粒子的开掘

能力方面考虑的, 所以局部极值的搜索能力得到极大的提高. 就测试函数 4 而言, 包含多个极值点, 该算法对粒子开拓能力的改进效果没有像改进开掘能力那样显著. 故而, 对于与测试函数 4 相类似的问题, 可以在算法运行的后期使用本文提出的改进算法, 效果将同以上改进其它五种测试函数一样具有显著效果. 限于篇幅,

作者将在以后的文中详细阐述。

表 4 固定时间长度仿真的结果

算法	测试函数 1			测试函数 2			测试函数 3			测试函数 4			测试函数 5			测试函数 6		
	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值	均值	方差	最小值
PSO-S	1708.56	206.35	1209.13	3044.33	2451.54	293.08	25.05	5.08	17.27	1.61	0.40	1.06	1.56	0.29	0.94	1.80	0.53	0.91
PSO-C	5.71	5.96	0.44	6.06	1.90	3.47	13.96	7.11	5.97	0.24	0.12	0.04	0.98	0.20	0.65	0.42	0.14	0.20
PSO-CIV	24.79	15.81	10.69	28.40	67.59	3.41	5.14	3.46	1.99	0.25	0.13	0.052	0.90	0.20	0.60	0.39	0.17	0.20
PSO-DIV	30.23	14.20	11.57	20.06	48.41	1.33	6.37	6.14	0.0015	0.24	0.14	0.067	0.92	0.37	0.50	0.42	0.13	0.30
PSO-MP	0	0	0	0.79	1.59	0	2.19	2.17	0	0.20	0.11	0.052	0.35	0.12	0.22	0.17	0.05	0.10

6 结论

本文提出了一种新型多步式位置可选择更新的粒子群算法,并对该算法的效能及复杂度等进行定性分析.采用六个典型函数进行测试表明该算法要优于现有的 4 种经典算法.该算法具有如下四方面特点:

(1)首次提出粒子群算法改进应当遵循算法最初提出时的简单实效性原则,在不增加粒子群算法的复杂性的基础上实现改进的思想,为粒子群改进算法增加了新的思想.

(2)受人类行为启发,首次提出多步式位置可选择更新的粒子群算法,充分利用速度更新公式的中间计算结果,实现粒子位置分三步更新,保存三步中的每一步累加值,最后选优为更新数据,达到了只增加很小的运算量就完成了进一步细化粒子的搜索空间的目的,提高了个体极值的更新率.

(3)与惯性权重等其它改进算法相比,本文算法无需依赖具体求解问题来调整设置过多的参数,不增加算法的时间复杂性,易操作,鲁棒性强、通用性好.

(4)在进行仿真测试时,本着科学研究的严谨性的态度,5 种算法不但相同参数取值相同,而且又采用了完全相同的随机初始种群,完全保持了各算法之间的可比较性,测试结果完全真实可信.

参考文献:

- [1] KENNEDY J E, EBERHART R C. Particle swarm optimization [A]. Proceedings of the IEEE International Conference on Neural Networks [C]. Perth, Australia: IEEE Press, 1995. 1942 - 1948.
- [2] Ali M M, KAALO P. Improved particle swarm algorithms for global optimization[J]. Applied Mathematics and Computation, 2008, 196: 578 - 593.
- [3] HENDILASS T. A combined swarm differential evolution algorithm for optimization problems [A]. Lecture Notes in Computer Science [C]. Berlin: Springer, 2001. 11 - 18.
- [4] SHI Y, EBERHART R C. Modified particle swarm optimizer [A]. Proceeding of the IEEE International Conference on Evolutionary Computation [C]. Piscataway NJ: IEEE Press, 1998. 69 - 73.

- [5] SHI Y, EBERHART R C. Parameter selection in particle swarm optimization [A]. Evolutionary Programming VII, Lecture Notes in Computer Science [C]. Berlin: Springer, 1998. 591 - 600.
- [6] LIU B, Wang L, JIN Y H, TANG F, HUANG D X. Improved particle swarm optimization combined with chaos [J]. Chaos, Solitons and Fractals, 2005, 25: 1261 - 1271.
- [7] SHI X H, LIANG Y C, LEE H P, LU C, WANG L M. An improved GA and a novel PSO-GA Hybrid algorithm [J]. Information Processing Letters, 2005, 93: 255 - 261.
- [8] DA Y, XINRUN G. An improved PSO-based ANN with simulated annealing technique [J]. Neurocomputing, 2005, 63: 527 - 533.
- [9] FOURIE P C, GROENWOLD A A. The particle swarm optimization algorithm in size and shape optimization [J]. Structural and Multidisciplinary Optimization, 2002, 23(4): 259 - 267.
- [10] CLERC M, KENNEDY J E. The particle swarm explosion, stability in a multidimensional complex space [J]. IEEE Transaction on Evolutionary Computation, 2002, 6(1): 58 - 73.
- [11] SCHUTTE J F, GROENWOLD A A. A study of global optimization using particle swarms [J]. Journal of Global Optimization, 2005, 31(1): 93 - 108.

作者简介:



高 芳 女, 1973 年 11 月出生于辽宁省, 分别于 1996 年、2002 年、2008 年获得哈尔滨工业大学计算机科学与技术学院工学学士、硕士和博士学位. 主要研究方向为群体智能算法、粒子群优化算法及应用.
E-mail: gaofang@hit.edu.cn



崔 刚 男, 1947 年 12 月出生于黑龙江省齐齐哈尔市. 现任哈工大计算机科学与技术学院教授, 博士生导师. 中国计算机学会高级会员, 中国计算机学会容错计算机专业委员会委员. 研究方向: 计算机体系结构、空间计算技术、高可信性计算技术、容错计算技术、嵌入计算技术、普适计算技术、移动计算技术、汽车计算技术等.