

文章编号:1672-3961(2010)05-0034-07

# 融合 PSO 算法思想的进化算法

刘建华<sup>1,2</sup>, 黄添强<sup>2</sup>, 严晓明<sup>2</sup>

(1. 福建工程学院计算机与信息系, 福建 福州 350007)

(2. 福建师范大学数学与计算机科学学院, 福建 福州 350007)

**摘要:** 粒子群算法(particle swarm optimization, PSO)是仿真于生物群体的社会行为的一种智能优化算法,其原始形式难以体现数学的直观性和本质性。然而,在简化算法原始模型的基础上,PSO 算法的理论分析得到其数学模型,并且说明了其是一个迭代进化系统。利用 PSO 算法的数学模型代替标准 PSO 算法速度及位置的迭代公式,并选择适当的参数,从而构造了一种新的进化算法。新的进化算法形式更能直接体现 PSO 算法的数学思想。经仿真试验表明,新的进化算法效果不亚于标准 PSO 算法,并且参数少且容易分析。

**关键字:** 粒子群算法;收敛性;进化算法;数学模型

中图分类号:TP18 文献标志码:A

## Evolutionary algorithm based on the idea of particle swarm optimization

LIU Jian-hua<sup>1,2</sup>, HUANG Tian-qiang<sup>2</sup>, YAN Xiao-ming<sup>2</sup>

(1. Department of Computer and Information, Fujian University of Technology, Fuzhou 350007, China)

(2. School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

**Abstract:** Particle swarm optimization (PSO) is an intelligence algorithm simulating the social behavior of a bird swarm or fish group. It is difficult for the original formula of PSO to show mathematical essence and principle. Using the simplified modal of PSO, the current theoretical analysis of PSO constructed a mathematical modal giving a clear essence of PSO from a mathematical view, this illustrated that the PSO was an iteration evolutionary system. Using the mathematical modal of PSO, a new evolutionary algorithm in which the velocity and location updating equation of PSO were replaced by the mathematical equations were developed. Some parameters of the new algorithm were discussed and properly selected. With selection of appropriate parameters, the performance of new evolutionary algorithm was not inferior to the standard PSO by simulation on benchmark functions. The new evolutionary algorithm was easy to understand and had mathematical meaning. Its parameters were fewer and easier to be analyzed than the standard PSO.

**Key words:** particle swarm optimization; convergence; evolutionary algorithm; mathematic modal

## 0 引言

PSO 算法是一种模拟群体智能的优化算法,其形式主要采用迭代递推方式。算法在迭代计算中考虑了各粒子之间的相互交互(全局点)及各维间的关系(函数适用值),所以粒子的迭代运动轨迹是一

个复杂系统。目前,对 PSO 进行算法理论分析是在简化系统的基础上,采用动态差分方程和线性系统理论进行的。这种分析方法虽然不完善,但它描述了 PSO 算法的思想原理,证明了 PSO 算法是一种迭代进化系统。基于 PSO 算法的思想原理,提出一种新的进化算法,并分析算法的参数选择和算法的收敛性。通过对基准函数的优化实验,对新算法与

收稿日期:2010-04-02

基金项目:福建省自然科学基金资助项目(2008J04004)

作者简介:刘建华(1967-)男,江西安福人,副教授,博士,主要研究方向为进化计算、数据挖掘等. E-mail: jhliu@fjnu.edu.cn

标准 PSO 进行了实验比较。

## 1 PSO 算法及其分析理论

PSO 算法是 R. Eberhart 和 J. Kennedy 于 1995 年开发的一种进化计算技术<sup>[1-2]</sup>,源于对鸟群捕食的行为研究。后来 Shi 等人引入惯性权重  $w$  来更好地控制收敛性 (convergence) 和探索性 (exploration)<sup>[3]</sup>,改进了 PSO 算法,形成了标准 PSO 算法。PSO 与其他进化算法相比,具有收敛较快、概念简单、实现容易等优点,且只有少数参数需要调整。目前,PSO 算法获得了很大的发展,并广泛地应用于许多领域。

假设算法搜索的解空间是  $D$  维的,粒子群中第  $i$  个粒子的位置用  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  表示,第  $i$  个粒子的速度表示为  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 。第  $i$  个粒子当前搜索到的最优位置为  $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ ,整个粒子群当前搜索到的最优位置为  $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 。对于每一个粒子,其第  $d$  维 ( $1 \leq d \leq D$ ) 根据下式变化:

$$v_{id} = wv_{id} + c_1 \text{rand}() (p_{id} - x_{id}) + c_2 \text{rand}() (p_{gd} - x_{id}), \quad (1)$$

$$x_{id} = x_{id} + v_{id}, \quad (2)$$

其中加速常数  $c_1$  和  $c_2$  是两个非负值,rand() 是在范围  $[0, 1]$  内取值的随机函数。 $w$  为惯性权重,它能平衡算法的全局搜索能力和局部搜索能力。

文献[3]进一步提出自适应调整的策略,即随着迭代的进行,线性减少  $w$  的值。这使得算法在迭代初期探索能力较强,可以不断搜索新的区域,然后算法的收敛能力逐渐增强,最后算法在可能的最优解周围精细搜索。这是当前广泛运用的标准 PSO 算法模型。

对于 PSO 算法单个粒子的运动轨迹分析,研究者为了获得数学上的研究模型,常常对原始 PSO 算法模型作一些简化的假定:粒子个体之间相互独立、算法最优解停止固定不变并且算法的随机性去掉。因此,最后的数学模型是近似的,并且需要试验验证。

1998 年, Ender Ozcan 和 C. K. Mohan 第一个对 PSO 算法作了理论分析,其分析方法是在简化的假定上进行<sup>[4-5]</sup>: 单个粒子的行为独立、在一维空间进行分析、去掉了算法随机性并且没有了最优解的动态进化,同时粒子个体最优位置 ( $p_i$ ) 和群体最优位置 ( $p_g$ ) 被假定为一,算法没有权重  $w$ , 也没速度限制。后来,同样的假定被应用到类似的分析中,这些

分析方法得出的结论为粒子群是一个离散线性系统,粒子状态的动态性通过求解状态转移矩阵的特征值来确定<sup>[6-8]</sup>,如果特征根的量值小于 1,则粒子将收敛于平衡点。因为系统的特征值是系统参数的函数,这种模型能建议算法的参数设置方法,以便算法能够保证收敛。M. Clerc 在简化算法的模型上,对单个粒子建立一个差分动态方程<sup>[9]</sup>,分析了参数与单个粒子收敛之间的关系。Trelea I. C. 建立单个粒子的差分方程<sup>[10]</sup>,在收敛条件下分析参数的选择范围。

Fan van den Bergh 在类似的简化模型<sup>[11]</sup>中,利用线性系统理论分析了粒子的收敛性,并得到粒子轨迹的模型。其分析得出,在参数的适当范围内,算法的粒子会收敛到一个固定点,这个点为粒子群体的最优位置与个体最优位置的加权平均点。Fan van den Bergh 在文献[11]对分析所得的结论作了比较详细推导。为了本文算法的构造,下面对其结论作简要叙述。

设  $t$  为 PSO 算法迭代运算的代数,令  $p(t)$  和  $p_g(t)$  分别为 PSO 算法第  $t$  代个体最优位置和群体最优位置。假定标准 PSO 算法的  $p(t)$  和  $p_g(t)$  不变,即  $p = p(t)$  和  $p_g = p_g(t)$ ; 令  $\phi_1 = c_1 \text{rand}()$ ,  $\phi_2 = c_2 \text{rand}()$  且  $\phi_1$  和  $\phi_2$  为常数。

根据式(1)和式(2)利用线性差分方程推得下列式子<sup>[11]</sup>:

$$x(t) = k_1 + k_2 \alpha^t + k_3 \beta^t, \quad (3)$$

其中:

$$k_1 = \frac{\phi_1 p + \phi_2 p_g}{\phi_1 + \phi_2}, \quad (4)$$

$$\alpha = \frac{1 + w - \phi_1 - \phi_2 + \gamma}{2}, \quad (5)$$

$$\beta = \frac{1 + w - \phi_1 - \phi_2 - \gamma}{2}, \quad (6)$$

$$\gamma = \sqrt{(1 + w - \phi_1 - \phi_2)^2 - 4w}, \quad (7)$$

$$k_2 = \frac{\beta(x(0) - x(1)) - x(1) + x(2)}{\gamma(\alpha - 1)}, \quad (8)$$

$$k_3 = \frac{\alpha(x(1) - x(0)) + x(1) - x(2)}{\gamma(\beta - 1)}, \quad (9)$$

$$x(2) = (1 + w - \phi_1 - \phi_2)x(1) - wx(0) + \phi_1 y_p + \phi_2 y_g. \quad (10)$$

若  $\phi_1$ 、 $\phi_2$ 、 $p$  和  $p_g$  是固定不变的常数且  $(1 + w - \phi_1 - \phi_2) < 4w$ , 则粒子位置  $x(t)$  的轨迹按下式收敛:

$$\lim_{t \rightarrow \infty} x(t) = (1 - \lambda)p + \lambda p_g, \lambda \in (0, 1), \quad (11)$$

其中  $\lambda = \frac{\phi_1}{\phi_1 + \phi_2}$ 。

## 2 新算法的模型

根据上节的简述,粒子群算法的数学模型是由式(3)、(4)、(5)和式(6)4个式子组成,而式(3)根据式(1)和式(2)推导出来的。虽然是在简化的条件下推得的,但式(3)可以认为是标准 PSO 算法中式(1)和式(2)的等价形式。因此,如果用式(3)代替标准 PSO 算法式(1)和式(2),并据此对算法的参数进行设置,则得到一种新的算法模型。但是,新算法模型必须考虑标准 PSO 算法的随机因素。所以,下面设计一种新的进化算法,用式(3)作为新进化算法的进化公式,代替标准 PSO 算法的速度更新式(1)和位置更新式(2),即有

$$x(t) = k_1 + k_2\alpha^t + k_3\beta^t,$$

这里  $k_1, k_2, k_3$  及  $\alpha, \beta$  都是参数,其意义见式(4)到式(9),而  $t$  是算法迭代运算的代数,即为算法的因变量。所以,新算法的位置是随进化代数  $t$  而变化。

根据式(4),可得:

$$k_1 = \frac{\phi_1 p + \phi_2 p_g}{\phi_1 + \phi_2} = \frac{\phi_1}{\phi_1 + \phi_2} p + \frac{\phi_2}{\phi_1 + \phi_2} p_g, \quad (12)$$

因此,当  $\phi_1, \phi_2$  是随机因素时,即有  $\phi_1 = c_1 \text{rand}()$ ,  $\phi_2 = c_2 \text{rand}()$ , 则  $k_1$  为如下形式:

$$k_1 = \text{rand}()p + (1 - \text{rand}())p_g. \quad (13)$$

式(13)说明  $k_1$  在粒子自身最优位置  $p$  和群体最优位置  $p_g$  之间线段上随机采样。把式(13)式代入式(3),则得粒子位置  $x(t)$  轨迹变化式为

$$x(t) = \text{rand}()p + (1 - \text{rand}())p_g + k_2\alpha^t + k_3\beta^t, \quad (14)$$

同样,新进化算法的粒子自身最优位置  $p$  和群体最优位置  $p_g$  必须迭代动态变化,令其分别为  $p(t)$  和  $p_g(t)$ 。它们的更新方法分别为如下两式:

$$\text{If } f(x(t)) < f(p(t)), \text{ then } p(t) = x(t), \quad (15)$$

$$\text{If } f(p(t)) < f(p_g(t)), \text{ then } p_g(t) = p(t), \quad (16)$$

故,式(14)变为如下形式:

$$x(t) = \text{rand}()p(t) + (1 - \text{rand}())p_g(t) + k_2\alpha^t + k_3\beta^t, \quad (17)$$

在式(17)中,粒子位置迭代变化的因变量为迭代步数  $t$ ,而  $k_2, k_3, \alpha$  和  $\beta$  都是参数,其取值将在下一节讨论分析。这样,标准 PSO 算法变为新的进化算法模型,其算法流程描述如下:

设粒子规模数为  $m$ ,最大迭代数为  $t_{\max}$ 。

**Step 1** 初始化:随机产生  $n$  个粒子位置;

**Step 2** 评价每个粒子的适应值;

**Step 3** 根据式(15)确定迄今为止每个粒子找到的最好位置  $p_i$ ;

**Step 4** 根据式(16)确定迄今为止整个群体找到的最好位置  $p_g$ ;

**Step 5** 根据式(17)调整粒子的位置  $x(t)$ ;

**Step 6** 若迭代条件没有满足且未达最大迭代步数  $t_{\max}$ ,则转向 Step2;

与标准 PSO 算法不同,新算法流程的 Step5 发生了变化,其粒子位置更新公式为式(17),不同于标准 PSO 算法的式(2),并且新算法没有了速度更新公式(1)。

## 3 参数设置与分析

根据式(17),新的进化算法需要设置参数  $k_2, k_3$  及  $\alpha, \beta$ 。为了使新进化算法尽量符合标准 PSO 算法的性能,新进化算法的参数设置需要考虑标准 PSO 的参数对它们的影响。

### 3.1 参数 $\alpha, \beta$ 的设置

根据式(5)和式(6),参数  $\alpha, \beta$  受到标准 PSO 算法的参数  $w, \phi_1$  和  $\phi_2$  的影响,而  $\phi_1$  和  $\phi_2$  是受随机变量影响,因为  $\phi_1 = c_1 \text{rand}()$ ,  $\phi_2 = c_2 \text{rand}()$ , 这里  $\phi_1$  和  $\phi_2$  在  $[0, c_i]$  ( $i=1, 2$ ) 区间进行随机均匀采样。在标准 PSO 算法中,权重  $w$  是一个随迭代而递减的参数。根据式(7),参数  $\alpha$  和  $\beta$  存在复数值的情况,但根据式(1)和式(2),标准 PSO 算法实际运行时,  $\alpha, \beta$  的值不应该存在复数。这是由于迭代时间  $t$  是整数,所以参数  $\alpha, \beta$  的值实际是实数,不可能为复数。

为了分析参数  $\alpha, \beta$  取值的范围,分析式(5)和式(6)两个式子的参数  $\alpha, \beta$  随权重  $w$  变化。由于这个两个式子很复杂,并且带有随机量  $\text{rand}()$ , 因此采用图形方式来演示其变化。利用 matlab, 演示参数  $\alpha, \beta$  随权重  $w$  的变化,其结果见图 1 和图 2。

根据图 1 和图 2 的参数  $\alpha, \beta$  随权重  $w$  的变化,当权重  $w$  从右向左变小时,参数  $\alpha, \beta$  的值会随机性地变小;当权重  $w < 1$  时,参数  $\alpha, \beta$  的值也小于 1。因此,当权重小于 1 时,式(3)收敛。这两个图形形象地说明标准 PSO 算法收敛性的要求。

根据图 1 和图 2,当权重在 0.4 到 1.4 变化时,参数  $\alpha, \beta$  的值大概在 0.4 到 0.6 之间变化波动,波动差异是  $[0, 0.8]$ 。由于标准 PSO 算法的权重变化一般从 1.4 到 0.4 的变化比较合适,因此,参数  $\alpha, \beta$  的取值设计为

$$0.8 \times \text{rand}() + w. \quad (18)$$

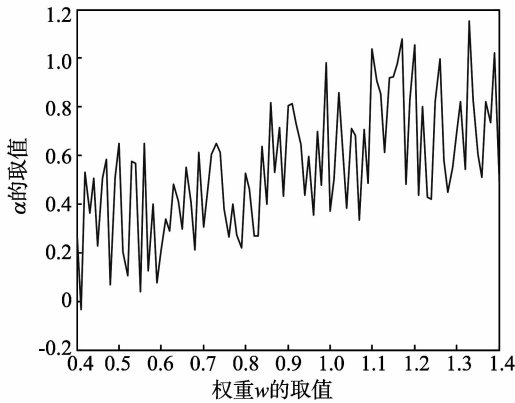


图1 参数  $\alpha$  随权重  $w$  变化

Fig.1 Variation of parameter  $\alpha$  with weight  $w$

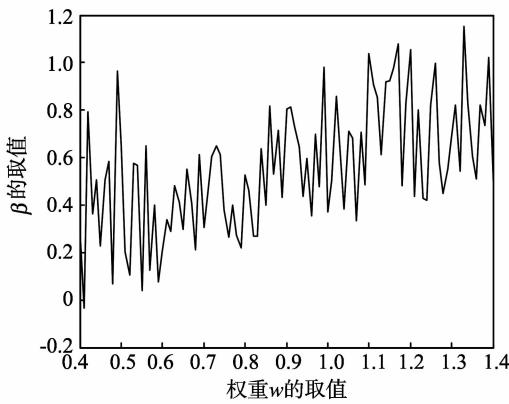


图2 参数  $\beta$  随权重  $w$  变化

Fig.2 Variation of parameter  $\beta$  with weight  $w$

这里,  $w$  随迭代时间步数  $t$  线性递减, 开始其值取最大值, 最后为最小值。根据图 1 和图 2, 新的进化算法的权重  $w$  从 0.4 递减变化到 0 比较合适。如果  $w$  取 0.4 到 0 的值, 则图 3 演示式 (18)。根据图 3, 式 (18) 的图形变化与图 1 基本吻合, 但要求参数  $w$  递减, 并带有随机数  $\text{rand}()$ 。递减目的是: 在早期时, 式 (18) 的值偏大; 在晚期时, 式 (18) 的值偏小。使早期算法具有发散性, 提高全局搜索能力。晚期式 (18) 的值偏小, 算法具有收敛性, 提高局部探测能力。

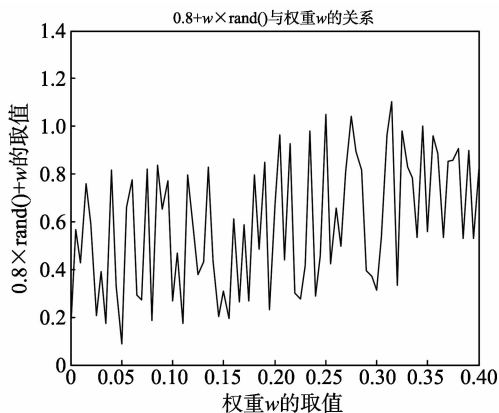


图3 式 (18) 取值 0 - 0.4 动态变化

Fig.3 Dynamical variation of value of formula (18)

参数  $\alpha, \beta$  的取值原则是两个参数的绝对值早期偏大, 为大于 1 的值, 而晚期为小于 1 的值, 并且带有随机性。这样的取值方式容易控制算法的收敛性, 也能更好地控制算法全局探测性与局部探索性的平衡。

### 3.2 参数 $k_2, k_3$ 的设置

同样, 为了确定参数  $k_2, k_3$ , 这小节分析式 (8) 和式 (9)。与参数  $\alpha, \beta$  的设置类似, 参数  $k_2, k_3$  的设置受标准 PSO 算法权重  $w$  影响, 并且还受粒子初始的三代位置  $x(0), x(1)$  和  $x(2)$  值的影响。根据式 (8) 和式 (9), 用 matlab 图形分析参数  $k_2, k_3$  与标准 PSO 算法权重  $w$  的关系。但还需考虑粒子的初始三代位置, 第一代位置  $x(0)$  是在搜索空间随机采样, 而第二代位置  $x(1)$  和第三代位置  $x(2)$  在标准 PSO 算法中受初始位置  $x(0)$  和初始速度  $v(0)$  的影响。这里为了考虑简单, 三代初始位置都认为在搜索空间中随机采样。在进化代数较多时, 这样的考虑是比较合理。

当初始位置  $x(0) = 5, x(1) = 5$  和  $x(2) = 5$  时, 图 4 演示了式 (8) 和式 (9) 中  $k_2, k_3$  与权重  $w$  的关系; 而当初始位置  $x(0) = -5, x(1) = -12$  和  $x(2) = -60$  时, 图 5 演示式 (8) 和式 (9) 中  $k_2, k_3$  与权重  $w$  的关系。4 个图形表现出如下两个特点: 第 1,  $k_2, k_3$  的值基本是在 0 值上下波动, 其值也可能为负, 也有可能为正。第 2,  $k_2, k_3$  存在突然巨变的情况, 绝对值达到上千。第 2 个特点是由于在推导式 (3) 时没有考虑标准 PSO 算法的速度限制值  $v_{\max}$ , 因此会存在巨变的现象。如果粒子速度值或粒子位置值被限制,  $k_2, k_3$  的值也将受到限制。这样, 粒子位置不可能超出搜索区域范围。

结合以上的两点特征,  $k_2, k_3$  的值是应该在 0 值的上下区域波动, 因此把  $k_2, k_3$  的形式改为如下形式:

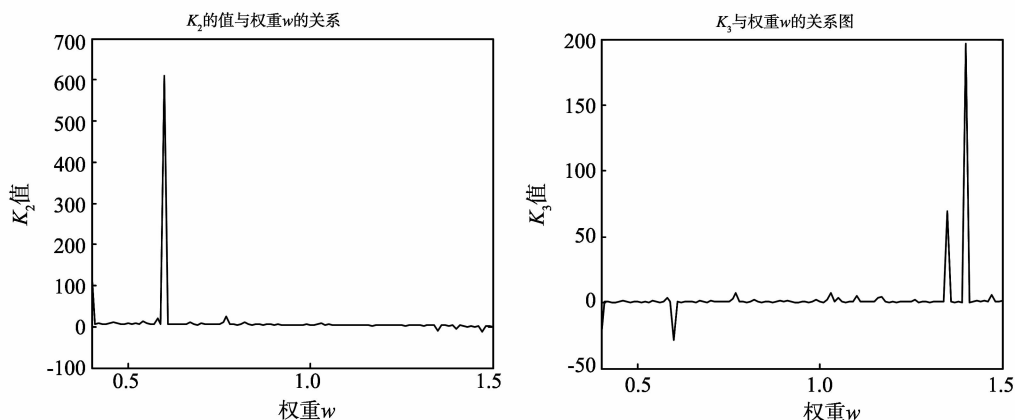
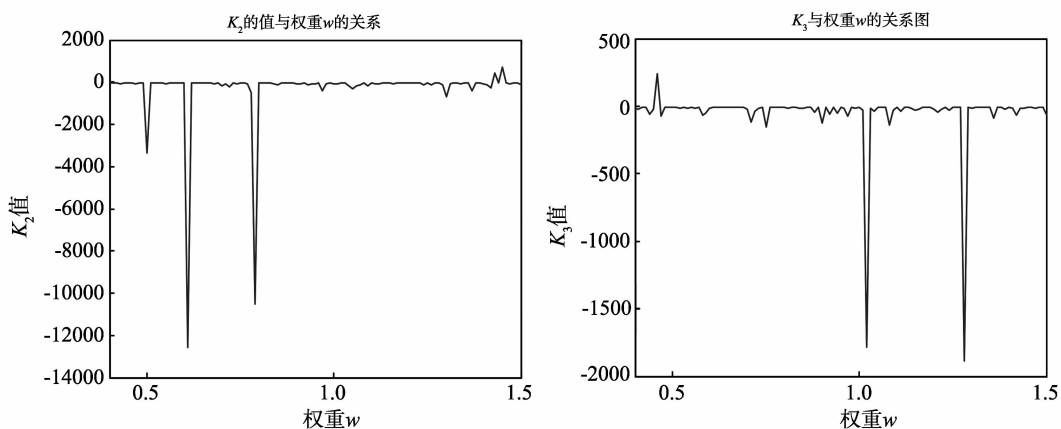
$$\pm c \times \text{rand}(), \quad (19)$$

这里,  $c$  是一个参数,  $\text{rand}()$  为  $[0, 1]$  区间的随机均匀采样数。

把式 (18) 和式 (19) 代入式 (17), 则得到如下式子:

$$x(t) = \text{rand}()p(t) + (1 - \text{rand}())p_g(t) + (-c \cdot \text{rand}()) \cdot (0.8 \text{rand}() + w)^t + (c \cdot \text{rand}()) \cdot (0.8 \text{rand}() + w)^t, \quad (20)$$

式 (20) 只要确定参数  $c$  和  $w$  即可, 其取值将在第 5 节的实验中分析讨论。

图4 初始位置  $x(0) = 5$ ,  $x(1) = 2$  和  $x(2) = 2$ Fig. 4 Initial position  $x(0) = 5$ ,  $x(1) = 2$  and  $x(2) = 2$ 图5 初始位置  $x(0) = -5$ ,  $x(1) = -12$  和  $x(2) = -60$ Fig. 5 Initial position  $x(0) = -5$ ,  $x(1) = -12$  and  $x(2) = -60$ 

### 3.3 初始化问题

随机性智能优化算法最终结果与初始值有关。本文算法由于没有初始速度,只需对粒子位置及每个个体最优位置与全局最优位置进行初始化。初始的方法是粒子的位置在搜索空间随机采样,而个体最优位置取粒子的初始位置,全局最优位置为整个粒子群初始位置中的最好位置。

### 3.4 算法的分析

根据式(20),本文算法收敛条件为式(18)的绝对值小于1,而这个式子的值由参数  $w$  来决定,只要选择合适参数,本文算法就会收敛于  $\text{rand}()p + (1 - \text{rand}())p_g$ ,而  $\text{rand}()p + (1 - \text{rand}())p_g$  的值由下式更新:

$$\text{if } f(x(t)) < f(p), \text{ then } p = x(t).$$

当  $x(t)$  跑到函数  $f(x)$  凸性区域时,会出现  $p \rightarrow p_g$ 。因此,算法最终会有  $x(t) \rightarrow p_g$ 。这个分析方法与文献[12]的分析说明是一致的。

进化算法早期必须具有全局探测性,晚期具有局部探索性。新算法能通过设置参数  $w$  的值来达到这个要求。因为,参数  $w$  的值越大,算法的全局探测能力越强;而参数  $w$  的值越小,算法局部探索

能力越强。所以,本算法也采用参数  $w$  随进化代数递减的方法,即  $w$  取值为  $0.8 \rightarrow 0.2$ 。算法早期处于发散状态,而晚期处于收敛状态。权重  $w$  的最小值  $0.2$  可使  $0.8 \times \text{rand}() + w$  式小于1,可以保持算法收敛。在算法实验运行中,进化代数一般会大于500,而  $(0.9)^{500} = 1.3221e-023$  是足够小的数。事实上,权重  $w$  的最小值可以根据算法的最大进化代数动态调整,算法的最大进化代数越多,权重  $w$  的最小值应该越大,但需要保持小于1,这样,可以避免算法早熟收敛。

## 4 实验与分析

为了验证本算法的有效性,选定5个基准测试函数(见表1)进行实验,用标准PSO算法和新算法分别对其进行优化计算,并进行比较。算法的参数设置如下:新算法权重  $w$  最大值为0.8,最小值为0.2;而标准PSO算法的权重  $w$  最大值为1.2,而最小值为0.4,两个算法的权重都随进化迭代而线性递减。参数  $c_1$ 、 $c_2$  都为1.8。种群数和进化代数根据各个测试的函数不同而设置不同的值,见表2~6。

表1 5个测试函数  
Table 1 Five test functions

函数名	表达式	维数	初值范围	目标值
Schaffer	$f_1(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001 * (x_1^2 + x_2^2))^2} + 0.5$	2	$[-5, 12, 5, 12]^2$	$10^{-3}$
Rastrigin	$f_2(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$n$	$[-5, 12, 5, 12]^n$	$10^2$
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$n$	$[-100, 100]^n$	$10^{-1}$
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$n$	$[-100, 100]^n$	$10^{-2}$
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$n$	$[-30, 30]^n$	$10^2$

表2 两种算法对 Schaffer's 函数 2 维的优化结果  
Table 2 Optimization result of function Schaffer by two algorithms

种群数	进化代数	计算类型	PSO	NPSO
		收敛次数	31	31
20	1 000	平均最优值	0.006 7	0.006 4
		平均收敛代数	873.580 6	580.354 8
		收敛次数	54	53
40	500	平均最优值	0.004 5	0.004 3
		平均收敛代数	435.648 1	286.018 9
		收敛次数	61	69
80	250	平均最优值	0.004 0	0.003 3
		平均收敛代数	182.983 6	141.159 4

表5 两种算法对 Rosenbrock 优化的结果  
Table 5 Optimization results of function Rosenbrock by two algorithms

种群数	进化代数	维数	计算类型	PSO	NPSO
			收敛次数	81	80
20	1 000	10	平均最优值	111.875 2	207.881 6
			平均收敛代数	497.012 3	869.250 0
			收敛次数	77	73
40	1 000	20	平均最优值	157.707 9	162.55
			平均收敛代数	589.649 4	443.458 3
			收敛次数	74	79
80	1 000	30	平均最优值	134.589 3	112.568 4
			平均收敛代数	646.378 4	651.782 6

表3 两种算法对 Rastrigin 函数 30 维的优化结果  
Table 3 Optimization results of function Rastrigin by two algorithms

种群数	进化代数	计算类型	PSO	NPSO
		收敛次数	72	86
20	1 000	平均最优值	102.562 7	98.564 6
		平均收敛代数	678.208 3	961.779 1
		收敛次数	84	83
40	500	平均最优值	99.906 6	99.363 1
		平均收敛代数	379.202 4	484.192 8
		收敛次数	83	69
80	250	平均最优值	99.617 0	102.986 4
		平均收敛代数	223.120 5	244.724 1

表6 两种算法对 Griewank 优化的结果  
Table 6 Optimization results of function Griewank by two algorithms

种群数	进化代数	维数	计算类型	PSO	NPSO
			收敛次数	100	100
20	1 000	10	平均最优值	0.006 7	0.006 4
			平均收敛代数	950.500 0	980.354 8
			收敛次数	99	94
40	1 000	20	平均最优值	0.091 5	0.092 5
			平均收敛代数	529.434 3	961.468 1
			收敛次数	100	99
80	1 000	30	平均最优值	0.004 1	0.003 3
			平均收敛代数	941.600 0	951.782 6

表4 两种算法对 Sphere 优化的结果  
Table 4 Optimization results of function Sphere by two algorithms

种群数	进化代数	维数	计算类型	PSO	NPSO
			收敛次数	100	99
20	1 000	10	平均最优值	0.008 3	0.006 4
			平均收敛代数	490.500 0	580.354 8
			收敛次数	100	93
40	1 000	20	平均最优值	0.008 9	0.009 3
			平均收敛代数	574.870 0	686.018 9
			收敛次数	100	95
80	1 000	30	平均最优值	0.009 1	0.009 9
			平均收敛代数	632.090 0	691.782 6

为了与标准 PSO 算法比较,每个算法运行 100 次,然后计算每个算法运行 100 次的收敛次数,平均最优值及平均收敛代数(见表 2~6)。

表 2 和表 3 是两个多峰函数的实验结果。表中数据说明新进化算法的优化效果不逊于标准 PSO 算法,因此完全可以看作标准 PSO 算法的等价算法。新算法与标准 PSO 算法相比,在种群粒子数越多的情况下,新进化算法效果更好。根据两个表的数据,每次实验都满足于:种群数 × 进化代数 = 20 000。这样的实验条件可以保持每次实验的时间

性不变。这个效果说明新算法更容易收敛,其收敛偏快,更易陷入局部最优。

若最大进化代数 为 250, 则  $(0.9)^{250} = 3.6360e-012$ , 这数值非常小, 足以说明算法收敛。由于更多粒子数有利于早期提高全局探索性, 而且本算法的局部收敛能偏强, 全局搜索能力偏弱, 因此需要增加粒子数来增加全局探索能力, 同时进化代数减少(为 250 代), 这样算法的时间性不变。

表 4 到表 6 显示了 3 个函数用标准 PSO 和新算法分别在不同维数计算的结果。从计算出的数据可以看出, 新算法的效果基本上与标准算法相当。这再次验证本文新算法基本可以替代标准 PSO 算法, 但对算法参数的具体取值需要进一步地精确分析。

## 6 结论

根据 PSO 算法的分析理论, 提出一种新的进化算法模型, 并分析了新算法的参数。如果新算法选择适当的参数, 新算法效果与标准 PSO 相当。新算法具有一定独特性, 并且参数不仅比标准 PSO 算法少, 而且其参数取值的数学意义更明晰。新算法的数学含义是粒子在自身最优点和全局最优点之间线段采样, 再加上一个具有迭代收敛性的式子, 然后附上自身最优点和全局最优点的更新公式, 这种数学模型的收敛原理很容易理解。标准 PSO 是模拟自然群体现象而得到的随机智能优化算法, 而本文的新算法根据它的数学模型, 构造新的随机优化。本文的思想为从数学模型中构造优化算法提供新视角。

### 参考文献:

- [1] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory[C]//The 6<sup>th</sup> International Symposium on Micro Machine and Human Science. Nagoya, Japan: IEEE Service Center, 1995: 78-79.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proceedings IEEE International Conference

on Neural Networks. Piscataway, NJ: IEEE Service Center, 1995:1942-1948.

- [3] SHI Y, EBERHART R C. A modified particle swarm optimizer[C]// Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Computer Society Press, 1998: 69-73.
- [4] OZCAN E, MOHAN C. Particle swarm optimization: surfing the waves[C]// Proceedings of 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Computer Society Press, 1999: 1939-1944.
- [5] OZCAN E, MOHAN C. Analysis of a simple particle swarm optimization system[J]. Intelligent Engineering Systems Through Artificial Neural Networks, 1998, 8: 253-258.
- [6] BRITS R, ENGELHRECHT A P, FAN ven den bergh. Locating multiple optimal using particle swarm optimization[J]. Applied Mathematics and Computation, 2007, 189: 1859-1883.
- [7] JACO F S, ALBERT A G. A study of global optimization using particle swarms[J]. Journal of Global Optimization, 2005, 31: 93-108.
- [8] VAN B F, ENGELBERCHT A P. A study of particle swarm optimization particle trajectories[J]. Information Sciences, 2006, 176(8): 937-971.
- [9] CLERC M, KENNEDY J. The particle swarm-explosion, stability and convergence in multidimensional complex space[J]. IEEE Transaction on Evolutionary Computation, 2002, 6(1): 58-73.
- [10] TRELEA I C. The particle swarm optimization algorithms: convergence analysis and parameter selection[J]. Information Processing Letters, 2003, 8(5): 317-325.
- [11] Van Den BERGH F. An analysis of particle swarm optimizer[D]. Pretoria, South Africa: University of Pretoria, 2002:78-143.
- [12] LIU Jianhua, FAN Xiaoping, QU Zhihua. The progressive analysis of particle swarm optimization[J]. Journal of Computational Information Systems, 2008, 4(6): 1829-1836.

(编辑:陈燕)