

# 均匀搜索粒子群算法的收敛性分析

吴晓军<sup>1,2</sup>, 李 峰<sup>2</sup>, 马 悦<sup>2</sup>, 辛云宏<sup>3</sup>

(1. 陕西师范大学现代教学技术教育部重点实验室, 陕西西安 710062;

2. 陕西师范大学计算机科学学院, 陕西西安 710062; 3. 陕西师范大学物理学与信息技术学院, 陕西西安 710062)

**摘 要:** 本文将均匀搜索粒子群算法(Uniform search Particle Swarm Optimization, 简称 UPSO)的位置更新公式转换为一个差分方程, 求解差分方程得到非递推的位置更新公式, 推导解的收敛条件并求出了 UPSO 对学习系数  $c$  及惯性系数  $w$  的收敛区域, 最后通过 6 个 Benchmark 函数仿真实验对收敛区域的正确性进行验证, 实验结果表明学习系数和惯性系数在收敛区域内时的 UPSO 收敛, 不在收敛区域外时 UPSO 发散.

**关键词:** 粒子群算法; 均匀搜索粒子群算法

**中图分类号:** TP301.6 **文献标识码:** A **文章编号:** 0372-2112 (2012)06-1115-06

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.06.008

## The Convergence Analysis of the Uniform Search Particle Swarm Optimization

WU Xiao-jun<sup>1,2</sup>, LI Feng<sup>2</sup>, MA Yue<sup>2</sup>, XING Yun-hong<sup>3</sup>

(1. Key Laboratory of Modern Teaching Technology, Ministry of Education, Shaanxi Normal University,

Xi'an, Shaanxi 710062, China; 2. School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi 710062, China;

3. School of Physics and Information Technology, Shaanxi Normal University, Xi'an, Shaanxi 710062, China)

**Abstract:** The uniform search particle swarm optimization (UPSO) algorithm formula was transformed into a differential equation. Solving the differential equation, we get a non-recurrence location update formula, and the UPSO's convergence region for learning coefficient  $c$  and inertia coefficient  $w$  were concluded by deducing the solution convergence conditions. Finally simulation experiments were provided on the selected location of the region of convergence by 6 Benchmark functions. Experimental results show that UPSO converges when the learning coefficient and inertial coefficient are in the convergence region and diverge outside convergence region.

**Key words:** particle swarm optimization; uniform search particle swarm optimization

## 1 引言

粒子群优化算法(Particle Swarm Optimization, 简称 PSO)是由 Eberhart 和 Kennedy 等人于 1995 年提出的一种基于种群的并行全局优化算法, 它源于对鸟群群体协作的捕食行为模拟, 通过种群间个体的协作, 引导整个群体向可能解的方向移动<sup>[1-3]</sup>. PSO 算法参数较少, 概念简单且容易实现, 可用于解决大量的非线性、不可微和多峰值等复杂的优化问题<sup>[1]</sup>, 近年来在科学和工程领域, 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获得了广泛的应用.

在基本 PSO 算法的基础上, 已经出现了多种 PSO

改进算法, 文献[4]通过实验给出了 PSO 算法参数的建议值, 文献[5]引入了惯性系数, 增强了算法的全局搜索能力, 使算法在没有最大速度限制的情况下依然表现出收敛的特性, 文献[6]对种群动态划分分子群, 在每个子群上使用 PSO 算法, 提高了 PSO 种群的多样性, 提高了算法效率, 文献[7]提出了协同 PSO (Cooperate Particle Swarm Optimization, CPSO), 克服了简单 PSO 算法易早熟、收敛慢的缺点, 使粒子更容易跳出局部极小点, 达到较高收敛精度. 文献[8]提出一种自适应粒子群优化算法 (Adaptive Particle Swarm Optimization, APSO), 该算法通过估计粒子群分布和适应度, 进而对进化过程中的状态进行实时估计, 并适应性选择较好的路径使粒子跳出可能

的局部最优,提高了收敛精度;还有一些学者从数学的角度对基本 PSO 算法的收敛性进行了分析,如文献[9]分析了简单粒子群优化算法的收敛性,指出它在满足收敛性的前提下种群多样性趋于减小,粒子失去继续搜索可行解的能力;并提出混沌粒子群优化算法提高种群的多样性和粒子搜索的遍历性.文献[10,11]用公式表示出不带惯性系数的简单粒子群算法的搜索轨迹,揭示了搜索过程中每个粒子在不同正弦波之间跳跃遴选更优解的实质,同时研究了不同参数条件下粒子群算法的收敛性,为简单粒子群算法的参数选取提供指导.文献[12]在文献[10]的基础上推导出带惯性系数的 PSO 算法的搜索轨迹,分析了参数选择对 PSO 算法收敛性的影响.文献[13]综述了收敛性分析的研究成果,证明了每个粒子都会收敛到一个稳定点的结论,并且分析了更新公式中的随机因子对收敛性的影响.文献[14]给出了 PSO 算法收敛性证明的模式,并证明了不带惯性系数的简单 PSO 算法位置更新公式本身不能保证算法收敛,而必须通过在计算时限制粒子的最大速度来达到收敛目的的缺陷,通过对更新公式的改进提出了一种保证收敛的算法(Guaranteed Convergence PSO, GCPSO).文献[15]将简单粒子群优化算法的收敛性问题转化成差分方程的特征解的收敛性问题,并对此进行分析、给出了收敛条件及实验证明,讨论了粒子群优化算法参数选取的基本原则,但文中对收敛条件的推导过程中忽略了特征值大于及等于 1 时不收敛的证明,其实验没有给出一群粒子总体表现分析.文献[16]应用概率统计的知识对目前对收敛性分析的主要方法中假设随机因子为常数得出的收敛区域的可靠性进行分析,并提出了用数学期望代替随机因子对带惯性系数的粒子群算法的收敛性分析的方法,文献[17]在收敛性分析的基础上,讨论了参数选择对粒子群算法的开发能力和探索能力的影响,提供了一种折衷的参数选择方案.文献[18]对粒子群算法的收敛性进行了分析,最后给出了一个参数选择的建议.

尽管以上研究成果极大的优化了 PSO 算法的性能,但均基于基本的 PSO 算法,而文献[19]通过分析搜索中心的概率分布,提出了搜索中心在两个极值间均匀分布的均匀搜索 PSO (Uniform Search Particle Swarm Optimization, UPSO) 算法,该算法较基本 PSO 在非均匀极值分布的函数优化中具有更高的效率.目前对基于 UPSO 算法的收敛性分析及算法优化的研究尚未展开.

## 2 均匀搜索粒子群算法原理

PSO 算法的核心是位置更新公式,它由 Kennedy 和 Eberhart<sup>[1]</sup>提出后经过 Shi 等人的改进<sup>[5]</sup>,形成了目前为广大学者普遍接受的形式.

设在  $D$  维空间上存在  $m$  个粒子,则在时间  $t$ , 粒子  $i$  在某分量上的位置按照公式(1)进行更新:

$$\begin{aligned} v_i(t+1) &= wv_i(t) + h_1r_1(t)[p_i(t) - x_i(t)] \\ &\quad + h_2r_2(t)[p_g(t) - x_i(t)] \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (1)$$

其中,  $i = 1, 2, \dots, m$ ;  $w$  为惯性系数,一般取  $(0, 1)$  之间的随机数;  $r_1$  和  $r_2$  是服从  $U(0, 1)$  分布的独立随机变量;学习因子  $h_1$  和  $h_2$  为非负常数;  $v_i \in [-v_{\max}, v_{\max}]$   $v_{\max}$  是由用户设定的约束速度的常数;  $p_i$  是粒子本身找到的历史最优解位置;  $p_g$  是种群目前找到的最优解位置;  $v_i$  是粒子的速度;  $x_i$  是粒子的当前位置. 经过研究, Eberhart 和 Shi 建议学习因子  $h_1, h_2$  取 1.49618, 惯性系数  $w$  取 0.7298<sup>[4]</sup>. 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. 若  $p_g$  为整个粒子群的最优位置,则上述算法也被称为全局版 PSO. 若  $p_g$  为第  $i$  个粒子的有限个相邻粒子搜索到的最优位置,则上述算法称为局部版 PSO, 一般情况下局部版 PSO 较全局版 PSO 具有更好的收敛稳定性<sup>[14]</sup>.

Kennedy 和 Clerc 在非随机性的条件下,通过理论分析证明,利用公式(1)进行位置更新的粒子,最终会收敛到一个稳定的位置  $p_c$ .

$$\lim_{t \rightarrow \infty} x(t) = p_c = \frac{h_1r_1}{h_1r_1 + h_2r_2}p_i + \frac{h_2r_2}{h_1r_1 + h_2r_2}p_g \quad (2)$$

$p_c$  在 PSO 算法迭代的过程中是启发式信息来源,对搜索中心  $p_c$  的最初分析源于文献[19]

设  $p_i \neq p_g$ , 且  $h_1r_1 + h_2r_2 > 0$ , 由式(2)可得:

$$\frac{p_c - p_i}{p_g - p_i} = \frac{h_2r_2(p_g - p_i)}{(h_1r_1 + h_2r_2)(p_g - p_i)} = \frac{h_2r_2}{h_1r_1 + h_2r_2} \quad (3)$$

计算  $p_c$  的联合概率密度函数式(4)

$$f_X(x) = \begin{cases} \frac{h_1/h_2}{2(1-x)^2}, & 0 \leq x \leq \frac{1}{1+h_1/h_2} \\ 0, & \text{其他} \\ \frac{1}{2h_1/h_2x^2}, & \frac{1}{1+h_1/h_2} \leq x \leq 1 \end{cases} \quad (4)$$

由式(4)可知,  $X$  并不服从均匀分布,这种不均匀的分布对于 PSO 的算法的效率存在一定的影响,在优化的过程中,尤其是普遍存在的多峰值函数优化中,导致更多的解具有相同的全局与局部最优,降低了粒子的多样性,影响了算法的稳定性.

基于以上分析,文献[19]提出了均匀搜索粒子群优化算法.均匀搜索粒子群算法是一种搜索中心  $p_c$  在  $p_i, p_g$  间均匀分布的 PSO 算法.根据文献[19],均匀粒子群搜索算法的位置更新为式(5)

$$\begin{aligned} v_i(t+1) &= wv_i(t) + c[rp_i(t) + (1-r)p_g(t) - x_i(t)] \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (5)$$

其中  $r$  为  $0 \sim 1$  之间均匀分布的随机数,  $c$  为学习

系数.

显然,式(5)所描述的算法在  $p_i, p_g$  之间具有相同的概率密度. 设  $p_i \leq p_g$ , 则搜索中心的概率密度函数为公式(6)

$$f(x) = \begin{cases} 1, & p_i \leq x \leq p_g \\ 0, & \text{其他} \end{cases} \quad (6)$$

文献[19]以 UPSO 算法替代基本 PSO 算法, 得到其改进后的相应的 UPSO 算法. 如基于 HPSO 的 HUPSO 算法, 基于 DPSO 的 DUPSO 算法等进行了实验验证. 实验数据证明, 以 UPSO 为基础的算法能够改善算法的性能. 尤其在非均匀分布多峰值函数优化中具有更加优良的性能.

由于 UPSO 算法采用了不同于 PSO 算法的位置更新公式, 它减少了一个参数, 减少了一个随机变量, 因此使得算法设计更加简单, 更易于控制. 在实际应用中, 参数如何选取能够保证算法收敛需要进一步深入研究.

### 3 均匀搜索粒子群算法的收敛性分析

尽管  $v_i$  和  $x_i$  是多维变量, 但每维相互独立, 故对算法分析可以简化到一维进行. 为简化计算, 并且假设粒子本身所找到的最优解的位置和整个种群目前找到的最优解的位置不变, 记为  $p$  和  $p_g$ ,  $r$  和  $c$  为常数. 由式(5)可推出:

$$x(t+1) - (1+w-c)x(t) + wx(t-1) = crp(t) + c(1-r)p_g(t) \quad (7)$$

这是一个迭代式, 无法直接分析粒子在  $t$  时刻的位置  $x(t)$  与初始条件的关系, 但式(7)是一个二阶常系数非齐次差分方程, 通过解此差分方程可求得  $x(t)$  与初始条件的关系, 下面利用特征方程方法求解此差分方程<sup>[15]</sup>.

公式(7)的系数矩阵的特征方程为

$$(\lambda^2 - (1+w-c)\lambda + w) = 0 \quad (8)$$

通过特征方程式(8)可以得到系数矩阵的 2 个特征值

$$\lambda_1 = \frac{1+w-c + \sqrt{(1+w-c)^2 - 4w}}{2}$$

$$\lambda_2 = \frac{1+w-c - \sqrt{(1+w-c)^2 - 4w}}{2}$$

根据  $\Delta = (1+w-c)^2 - 4w$  与 0 的大小关系对式(7)分三种情况求解:

(1) 当  $\Delta < 0$  时,  $\lambda_{1,2} = \frac{1+w-c \pm j\sqrt{4w - (1+w-c)^2}}{2}$  此时可以得到非递推的位置更新表达式:

$$x(t) = k_0 + k_1\lambda_1^t + k_2\lambda_2^t \quad (9)$$

其中  $k_0, k_1, k_2$  为待定系数, 经计算得到:

$$k_0 = rp(t) + (1-r)p_g(t)$$

$$k_1 = \frac{\lambda_2 x(0) - x(1)}{\lambda_2 - \lambda_1}$$

$$k_2 = \frac{\lambda_1 x(0) - x(1)}{\lambda_1 - \lambda_2}$$

(2) 当  $\Delta = 0$  时,  $\lambda_1 = \lambda_2 = \frac{1+w-c}{2}$  得到非递推的位置更新表达式:

$$x(t) = k_0 + (k_1 + k_2)\lambda^t \quad (10)$$

其中  $k_0, k_1, k_2$  为待定系数, 经计算得到:

$$k_0 = rp(t) + (1-r)p_g(t)$$

$$k_1 = x(0)$$

$$k_2 = \frac{x(1) - x(0)}{\lambda} - x(0)$$

(3) 当  $\Delta > 0$  时,  $\lambda_{1,2} = \frac{1+w-c \pm \sqrt{(1+w-c)^2 - 4w}}{2}$  得到非递推的位置更新表达式

$$x(t) = k_0 + k_1\lambda_1^t + k_2\lambda_2^t \quad (11)$$

其中  $k_0, k_1, k_2$  为待定系数, 经计算得到:

$$k_0 = rp(t) + (1-r)p_g(t)$$

$$k_1 = \frac{\lambda_2 x(0) - x(1)}{\lambda_2 - \lambda_1}$$

$$k_2 = \frac{x(1) - x(0)\lambda_1}{\lambda_1 - \lambda_2}$$

为了求解式(7), 假设单个粒子的最优位置  $p(t)$  和种群的全局最优位置  $p_g(t)$  为不变的常数, 即  $k_0 = rp(t) + (1-r)p_g(t)$  为常数, 在粒子更新实际情况中, 如果  $p(t), p_g(t)$  没有变化则  $k_0$  的值是正确的, 但是当更新过程有发现新的最优解  $p'(t), p'_g(t)$  时,  $k_0 \neq rp'(t) + (1-r)p'_g(t) = k'_0$ , 因此需要用新得到的最优位置计算出  $k'_0$  的值并替换原来的  $k_0$ , 这样  $k'_0$  的值在下次最优位置改变之前都是正确的.

由式(9)~(11)可知, 如果要  $x(t)$  收敛, 需要所有特征值小于 1, 也就是  $\lambda_1$  和  $\lambda_2$  的绝对值必须小于 1, 这样粒子的位置按式(12)收敛<sup>[13]</sup>:

$$\lim_{t \rightarrow \infty} x(t) = rP_L + (1-r)P_g \quad (12)$$

下面推导  $x(t)$  的收敛条件<sup>[17]</sup>:

首先, 式(9)~(11)均可统一表示成三角形式:

$$x(t) = k_0 + k_1\lambda_1^t + k_2\lambda_2^t$$

$$= k_0 + k_1 \|\lambda_1\|^t (\cos(\theta_1 t) + j \sin(\theta_1 t))$$

$$+ k_2 \|\lambda_2\|^t (\cos(\theta_2 t) + j \sin(\theta_2 t))$$

其中  $\theta_1 = \arg(\lambda_1), \theta_2 = \arg(\lambda_2)$ , 因为  $\lambda_1$  和  $\lambda_2$  是式(7)的

特征根,所以  $\lambda_1$  与  $\lambda_2$  正交,即  $\theta_2 = -\theta_1$

数列  $\{x(t)\}$  的极限为:

$$\begin{aligned} \lim_{t \rightarrow \infty} x(t) &= \lim_{t \rightarrow \infty} (k_0 + k_1 \lambda_1^t + k_2 \lambda_2^t) \\ &= \lim_{t \rightarrow \infty} (k_0 + k_1 \|\lambda_1\|^t (\cos(\theta_1 t) + j \sin(\theta_1 t)) \\ &\quad + k_2 \|\lambda_1\|^t (\cos(\theta_1 t) - j \sin(\theta_1 t))) \\ &= \lim_{t \rightarrow \infty} (k_0 + (k_1 + k_2) \|\lambda_1\|^t \cos(\theta_1 t) \\ &\quad + j(k_1 - k_2) \|\lambda_1\|^t \sin(\theta_1 t)) \end{aligned}$$

当  $\lambda > 1$  时显然均匀搜索粒子群算法不收敛。

其中  $\lambda = 1$  时,  $\lim_{t \rightarrow \infty} \lambda^t = \lim_{t \rightarrow \infty} 1^t (\cos(\theta t) + j \sin(\theta t))$ , 此极限不存在, 即当  $\lambda = 1$  时算法不收敛

当  $\lambda < 1$  时需要根据  $\Delta$  的正负分类分析:

(1) 当  $(1+w-c)^2 - 4w < 0$  时:  $\lambda_1$  和  $\lambda_2$  为复数, 则

$$\begin{aligned} \|\lambda_1\| &= \|\lambda_2\| \\ &= \frac{1}{2} \sqrt{(1+w-c)^2 - (1+w-c)^2 + 4w} = w \end{aligned}$$

所以当  $w < 1$  并且  $(1+w-c)^2 - 4w < 0$  时, 特征值的绝对值小于 1, 所以, 当系统参数满足

$$\begin{cases} w < 1 \\ (1+w-c)^2 - 4w < 0 \end{cases} \quad \text{时, 式(9)收敛.}$$

(2) 当  $(1+w-c)^2 - 4w = 0$  时: 此时  $|\lambda| =$

$$\begin{aligned} \left| \frac{1+w-c}{2} \right| < 1, \text{ 由此可得当系统满足} \\ \begin{cases} (1+w-c)^2 = 4w \\ -3 < w-c < 1 \end{cases} \quad \text{时, 式(10)收敛.} \end{aligned}$$

(3) 当  $(1+w-c)^2 - 4w > 0$  时:  $\lambda_1$  和  $\lambda_2$  为实数, 则此时只要两个特征值的最大值小于 1, 即  $\max(\|\lambda_1\|, \|\lambda_2\|) < 1$ , 此时分三种情况讨论:

①  $1+w > c$

此时  $\lambda_1 > \lambda_2$ , 所以此时只要满足  $\lambda_1 < 1$ , 算法即收敛, 所以可以得到:

$$\frac{1+w-c+\sqrt{(1+w-c)^2-4w}}{2} < 1$$

从中可以推得  $c > 0$ , 所以当系统参数满足

$$\begin{cases} c > 0 \\ (1+w-c)^2 - 4w > 0 \\ 1+w > c \end{cases} \quad \text{时, 式(11)收敛.}$$

②  $1+w < c$

此时  $\lambda_1 < \lambda_2$ , 所以此时只要满足  $|\lambda_2| < 1$ , 算法即收敛, 所以可以得到:

$$\frac{c-1-w+\sqrt{(1+w-c)^2-4w}}{2} < 1$$

从中可以推得  $c < 2(w+1)$ , 所以当系统参数满足

$$\begin{cases} (1+w-c)^2 - 4w > 0 \\ 1+w < c < 2(w+1) \end{cases} \quad \text{时, 式(11)收敛.}$$

③  $1+w = c$

$$\begin{aligned} \text{此时 } |\lambda_1| = |\lambda_2| &= \frac{\sqrt{(1+w-c)^2-4w}}{2}, \text{ 此时需满足:} \\ \frac{\sqrt{(1+w-c)^2-4w}}{2} &< 1 \end{aligned}$$

从中可得  $-1 < w < 0$ , 所以当系统参数满足

$$\begin{cases} (1+w-c)^2 - 4w > 0 \\ 1+w = c \\ -1 < w < 0 \end{cases} \quad \text{时, 式(11)收敛.}$$

由文献[20]可知,  $w$  为粒子当前速度对粒子下一迭代周期速度的影响因子, 当  $w < 0$  时粒子的速度一直减少, 直到速度为 0, 所以  $w > 0$ . 综合以上条件:

$$\begin{cases} w < 1 \\ (1+w-c)^2 - 4w < 0 \end{cases} \quad (13)$$

$$\begin{cases} (1+w-c)^2 = 4w \\ -3 < w-c < 1 \end{cases} \quad (14)$$

$$\begin{cases} c > 0 \\ (1+w-c)^2 - 4w > 0 \\ 1+w > c \end{cases} \quad (15)$$

$$\begin{cases} (1+w-c)^2 - 4w > 0 \\ 1+w < c < 2(w+1) \end{cases} \quad (16)$$

$$\begin{cases} (1+w-c)^2 - 4w > 0 \\ 1+w = c \\ -1 < w < 0 \end{cases} \quad (17)$$

其中式(17)不满足  $w > 0$  的条件. 综合其他四个条件可以得到均匀粒子群算法位置收敛的参数条件:

$$\begin{cases} 0 < w < 1 \\ 0 < c < 4 \\ c < 2(w+1) \end{cases} \quad (18)$$

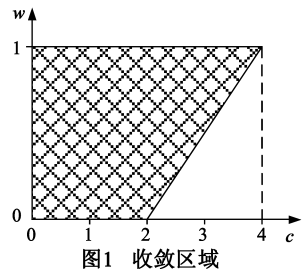


图1 收敛区域

收敛区域如图 1 所示。

## 4 仿真实验

### 4.1 实验设计

为了比较参数选取在第 3 部分求出的收敛区域内外对 UPSO 算法的收敛性的影响, 实验部分分  $\Delta < 0$ 、 $\Delta = 0$  和  $\Delta > 0$  三种情况

表 1 实验参数选择

参数 分组	收敛区域内		收敛区域外	
	w	c	w	c
$\Delta < 0$	0.81	0.1	4	2
$\Delta = 0$	0.36	0.16	1	4
$\Delta > 0$	0.09	0.4	0.5	5

进行实验, 每组分别选一组在收敛区域内和一组在收敛区域外的参数进行对比, 表 1 列出了算法实验参数的选择。

本文引入了 PSO 以及遗传算法 (Genetic Algorithm, 简称 GA) 常用的 6 个 Benchmark 优化函数进行实验, 其表达式如表 2 所示, 对每一个函数, 维数设置为 30, 粒

子规模为 100, 迭代 6000 次进行优化, 并独立运行 50 次, 计算在每一次迭代上最优解的平均值, 最后绘制最小值-迭代次数曲线.

4.2 实验结果

第一组( $\Delta < 0$ )实验, 对于  $w = 0.81, c = 0.1$ , 由图 1 可知参数(0.81, 0.1)在收敛区域内, 粒子群的最优位置与迭代次数关系在图 2 中用虚线表示, 由图 2 可知对 6 个 Benchmark 函数 UPSO 都迭代收敛; 对于  $w = 4, c = 2$ , 由图 1 可知参数(4, 2)在收敛区域外, 粒子群的最优位置与迭代次数关系在图 2 中用带星号的实线表示, 由图 2 可知对 6 个 Benchmark 函数 UPSO 都迭代发散.

第二组( $\Delta = 0$ )实验, 对于  $w = 0.36, c = 0.16$ , 由图 1 可知参数(0.36, 0.16)在收敛区域内, 粒子群的最优位置与迭代次数关系在图 2 中用实线表示, 由图 2 可知对 6 个 Benchmark 函数 UPSO 都迭代收敛; 对于  $w = 1, c = 4$ , 由图 1 可知参数(1, 4)在收敛区域外, 粒子群的最优位置与迭代次数关系在图 2 中用带点的

表 2 Benchmark 函数原型

Function	Name	Search space	Minimum
$f_1 = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	Griewank	$(-300, 300)^N$	0
$f_2 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	Rastrigin	$(-5.12, 5.12)^N$	0
$f_3 = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$	Tablet	$(-100, 100)^N$	0
$f_4 = \sum_{i=1}^n (100(x_{i+1} - x_i^2) + (x_i - 1)^2)$	Rosenbrock	$(-50, 50)^N$	0
$f_5 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	Quadric	$(-100, 100)^N$	0
$f_6 = \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} [\sin(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1]$	Schaffer	$(-100, 100)^N$	0

虚线表示, 由图 2 可知对 6 个 Benchmark 函数 UPSO 都迭代发散.

第三组( $\Delta > 0$ )实验, 对于  $w = 0.09, c = 0.4$ , 由图 1 可知参数(0.09, 0.4)在收敛区域内, 粒子群的最优位置与迭代次数关系在图 2 中用点划线表示, 由图 2 可知对 6 个 Benchmark 函数 UPSO 都迭代收敛; 对于  $w = 0.5, c = 5$ , 由图 1 可知参数(0.5, 5)在收敛区域外, 粒子群的最优位置与迭代次数关系在图 2 中用带点的实线表示, 由图 2 可知对 6 个 Benchmark 函数 UPSO 都迭代发散.

由上面的三组实验结果可以看出: 当选取收敛区域内的参数时, 随着迭代次数的增加, 粒子群的适应值逐渐减少, 越来越接近最优解, UPSO 迭代收敛; 当选取收敛区域外的参数时, 由于每个粒子的位置随着迭代次数的增大, 无法找到更小的适应值代替原来的适应值, 所以粒子群的适应值一直不变, UPSO 迭代发散. 这说明本文第 3 部分解出的收敛区域是正确的.

5 结束语

均匀搜索粒子群算法提出了搜索中心的概念, 引入了新的位置更新公式, 构造了一个搜索中心均匀分布于两个最优解之间的搜索算法, 在复杂空间上具有更好的全局搜索能力, 与基本粒子群算法相比, UPSO 算法简单且性能稳定, 并且减少了一个学习系数及独立随机变量.

本文对均匀搜索粒子群算法的位置更新公式进行了变换, 将其构造为一个二阶常系数非齐次差分方程, 在此基础上对粒子优化过程中的位置  $x(t)$  的收敛条件进行了分析, 得到了 UPSO 算法中学习系数  $c$  及惯性系数  $w$  的收敛区域. 最后通过 6 个 Benchmark 函数的优化实验, 验证了论文的研究结果.

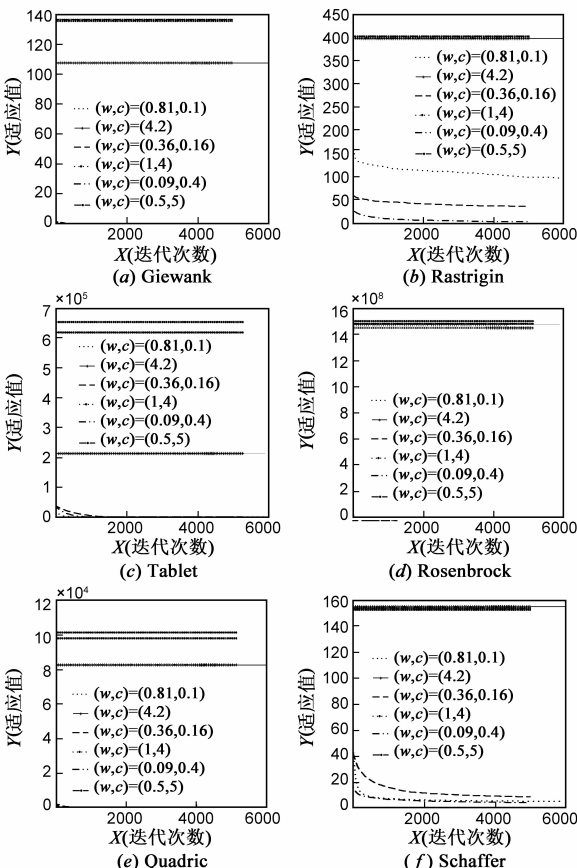


图 2 不同惯性系数  $w$  及学习系数  $c$  选取下 6 个 Benchmark 函数的最优位置与迭代次数关系

## 参考文献

- [1] J Kennedy, R C Eberhart. Particle swarm optimization [A]. Proceedings IEEE International Conference on Neural Networks [C]. Perth, Australia: IEEE Press, 1995. 1942 – 1948.
- [2] R C Eberhart, J Kennedy. A new optimizer using particle swarm theory [A]. Proceedings of the Sixth International Symposium on Micro Machine and Human Science [C]. Nagoya, Japan: IEEE Press, 1995. 39 – 43.
- [3] R C Eberhart, P Simpson, R Dobbins. Computational Intelligence PC Tools [M]. Boston: Academic Press Professional, 1996. 212 – 226.
- [4] Y Shi, R C Eberhart. Parameter selection in particle swarm adaptation [A]. Proc 7th Annual Conference on Evolutionary Programming [C]. Washington DC: IEEE Press, 1998. 591 – 600.
- [5] Y Shi, R C Eberhart. A modified particle swarm optimizer [A]. Process of 1998 IEEE International Conference on Evolutionary Computation [C]. Anchorage, AK: IEEE Press, 1998. 69 – 73.
- [6] Yan Jiang, Tiesong Hua, Chong Chao Huang. An improved particle swarm optimization algorithm [J]. Applied Mathematics and Computation, 2007, 193(1): 231 – 239.
- [7] F van den Bergh, A P Engelbrecht. A cooperative approach to particle swarm optimization [J]. IEEE Transactions On Evolutionary Computation, 2004, 8(3): 225 – 239.
- [8] ZH Zhan, J Zhang, Y Li, HSH Chung. Adaptive particle swarm optimization [J]. IEEE Transactions on Systems Man and Cybernetics, 2009, 39(6): 1362 – 1381.
- [9] 刘洪波, 王秀坤, 谭国真. 粒子群优化算法的收敛性分析及其混沌改进算法 [J]. 控制与决策, 2006, 21(6): 636 – 645.  
LIU Hong-bo, WANG Xiu-kun, TAN Guo-zhen. Convergence analysis of particle swarm optimization and its improved algorithm based on chaos [J]. Control and Decision, 2006, 21(6): 636 – 645. (in Chinese)
- [10] E Ozcan, C K Mohan. Analysis of a simple particle swarm optimization system [J]. Intelligent Engineering Systems through Artificial Neural Networks, 1998, 8: 253 – 258.
- [11] E Ozcan, C K Mohan. Particle swarm optimization: Surfing the waves [A]. Congress on Evolutionary Computation (CEC' 1999) [C]. Washington, DC, USA: IEEE Press, 1999. 1939 – 1944.
- [12] F van den Bergh. An Analysis of Particle Swarm Optimizers [D]. Pretoria, South Africa: Department of Computer Science, University of Pretoria, 2002.
- [13] F van den Bergh, A P Engelbrecht. A study of particle swarm optimization particle trajectories [J]. Information Science, 2006, 176(8): 937 – 971.
- [14] F van den Bergh, A P Engelbrecht. A convergence proof for the particle swarm optimiser [J]. Fundamenta Informaticae, 2010, 105(4): 341 – 374.
- [15] 高尚, 汤可宗, 蒋新姿, 杨静宇. 粒子群优化算法收敛性分析 [J]. 科学技术与工程, 2006, 6(12): 1625 – 1627.  
GAO Shang, TANG Ke-zong, JIANG Xin-zi, YANG Jing-yu. Convergence analysis of particle swarm optimization algorithm [J]. Science Technology and Engineering, 2006, 6(12): 1625 – 1627. (in Chinese)
- [16] R C Eberhart, Y Shi. Comparing inertia weights and constriction factors in particle swarm optimization [A]. Evolutionary Computation, 2000 Proceedings of the 2000 Congress on [C]. CA, USA: IEEE Press, 2000. 84 – 88.
- [17] RPoli. The sampling distribution of particle swarm optimisers and their stability [DB/OL]. <http://cswww.essex.ac.uk/technical-reports/2007/csm-465.pdf>, 2007.
- [18] IC Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection [J]. Information Processing Letters, 2003, 85(6): 317 – 325.
- [19] 吴晓军, 杨战中, 赵明. 均匀搜索粒子群算法 [J]. 电子学报, 2011, 39(6): 1261 – 1266.  
WU Xiao-jun, YANG Zhan-zhong, ZHAO Ming. A uniform searching particle swarm optimization algorithm [J]. Acta Electronica Sinica, 2011, 39(6): 1261 – 1266. (in Chinese)
- [20] M Clerc, J Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58 – 73.

## 作者简介



吴晓军(通讯作者) 男, 1970年生, 陕西凤翔人, 陕西师范大学计算机科学学院副教授。研究方向为模式识别与智能系统, 嵌入式系统。  
E-mail: wythe@snnu.edu.cn

李峰 男, 1986年生, 广西桂林市人, 陕西师范大学计算机科学学院硕士研究生, 研究方向为普适计算, 计算机系统结构。

马悦 女, 1988年生, 陕西省西安市人, 陕西师范大学计算机科学学院硕士研究生, 研究方向为智能控制与应用系统, 计算机应用技术。

辛云宏 男, 1967年生, 陕西蒲城人, 陕西师范大学物理与信息技术学院副教授, 研究方向为信号检测与处理。