第六章 PSO 算法的实验设计与参数选择

6.1 典型实验函数

为分析与检验微粒群算法的性能，目前该领域的研究人员一般采用下列一些 BenchMark 测试函数。

6.1.1 无约束优化测试函数

F1： $f_1(X)=\sum_{i=1}^{n} x_i^2$ $\qquad 100 \le x_i \le 100$

F2： $f_2(X)=\sum_{i=1}^{n} 100\left(x_{i+1}^2-x_i\right)+(1-x_i)^2$ $\qquad 30 \le x_i \le 30$

F1 与F2 是由K.A.DeJong提出的。F1 是著名的Sphere函数，单峰，在x_i=0 时达到极小值；F2 被称为Rosenbrock函数.，非凸、病态函数，在x_i=1 时达到极小值.

F3： $f_3(X)=\sum_{i=1}^{n}\left(x_i^2-ACOS\ (2\pi x_i)+A\right)$ $\qquad -5.12 \le x_i \le 5.12$

F3 被称为 Rastrigin 函数，多峰，在 $x_i=0(i=1\cdots n)$ 时达到全局极小点，在 $S=\{x_i \in (-5.12,5.12), i=1,2,\cdots n\}$ 范围内大约存在 10n 个局部极小点。

F4： $f_4(X)=\sum_{i=1}^{n} x_i^2 \Big/ 4000-\prod_{i=1}^{n} COS\ (x_i/\sqrt{i})+1$ $\qquad -60 \le x_i \le 600$

函数 F4 是由 Griewank 提出的。全局极小在 $x_i=0(i=1\cdots n)$ 时达到，局部极小值在 $x_i \approx \pm k \bullet \pi\sqrt{i},\quad i=1,2,\cdots,n;\quad k=0,1,2,\cdots,n$。

F5： $f_5(X)=0.5-\dfrac{(\sin^2\sqrt{x_1^2+x_2^2}-0.5)}{(1+0.001(x_1^2+x_2^2))^2}$ $\qquad -10 \le x_1, x_2 \le 100$

F5 是著名的 Schaffer 函数，由 J.D.Schaffer 提出，.全局极大点是(0,0),在距离全局极大点大约 3.14 的范围内存在无限多的次全局极大点，函数强烈震荡的性态使其很难于全局最优化。

F6： $f_6(X)=\sum_{i=1}^{5}[i\cos((i-1)\ x_1+i)]\sum_{j=1}^{5}[j\cos((j+1)x_2+j)]+(x_1+1.42513)^2+(x_2+0.80032)^2$ $\qquad -10 \le x_i \le 10$

LevyNo.5 函数(Levy 1981)，著名的两维测试函数，具有 760 个局部极值点,全局最优解为 x^*(-1.3.68,-1.4248)，最优值 $F(x^*)=-176.1375$。由于众多的局部极值点，使得各种优化算法很难搜索到全局最优解。

F7： $f_7(X)=\sum_{i=1}^{5}[i\cos((i-1)x_1+i)]\sum_{j=1}^{5}[j\cos((j-1)x_2+j]$ $-10\le x_i\le 10$

Levy No.3 函数(Levy 1981)，具有 760 个局部极小点和 18 个全局极小点，最优值 $F(x^*)=-176.542$。

F8： $f_8(X)=\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2$

Levy No.8 函数，其中 $y_i=1+\frac{x_i-1}{4}$，$x_i\in[-10,10]$，$i=1,2,3$。函数在 $x^*=(1,1,1)^T$ 处得到极小值 $F(x^*)=0$，同时约有 125 个局部极小值(Levy 1981)。

F9: $f_9(x)=\sum_{j=1}^{4}\begin{cases}0.15[z_j-0.05Sign(z_j)]^2 d_j, & if \quad |x_j-z_j|<0.05\\ d_j x_j^2 & else\end{cases}$

$x_j\in[-1000,1000]$，$(d1，d2，d3，d4)=(1,1000,10,100)$

其中，$Zj=(\left|\frac{x_j}{0.2}\right|+0.49999)\bullet Sgn(x_j)\times 0.2$

F9 为 Corana 函数[Corana 1987]，很难极小化。

F10： $f_{10}(x)=-13+x_1+((5-x_2)x_2-2)x_2]^2$
$+[-29+x_1+((x_2+1)x_2-14)x_2]^2$ v

Freudenstein-Roth 函数[More 1981]，极小值 $F(x^*)=0$，$x^*=(5,4)^T$

F11： $f_{11}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$
$\times[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$

其中 $x_i\in[-5.12,+5.12)$。此为 Goldstern-Price 函数[More 1981]，min$f(x)$=3, $x^*=(0,-1)^T$

6.1.2 多目标优化测试函数

F1： $f_1=\frac{1}{n}\sum_{i=1}^{n}x_i^2$ $f_2=\frac{1}{n}\sum_{i=1}^{n}(x_i-2)^2$

F2： $f_1=x_1$ $g=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$ $f_2=g(1-\sqrt{f_1/g})$

F3： $f_1=x_1$ $g=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$ $f_2=g(1-(f_1/g)^2)$

F4： $f_1=x_1$ $g=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$ $f_2=g(1-\sqrt[4]{f_1/g}-(f_1/g)^4)$

F5： $f_1 = x_1 \quad g = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \qquad f_2 = g(1 - \sqrt{f_1 / g} - (f_1 / g)\sin(10\pi f_1)$

上述五个测试问题出自文献[Knowles and Corne 2000; Zitzler 2000]，虽然看似简单，但却代表了各种不同 Pareto 前沿特征的 MO 问题。其中 F1 具有均匀分布的 Pareto 前沿，且为凸集；F2 的 Pareto 前沿为非均匀分布的凸集；F3 的 Pareto 前沿为凹的；F4 与 F5 的 Pareto 前沿形态较为复杂，前者的 Pareto 前沿既非凸、也非凹，而后者的 Pareto 前沿则包含了几个独立的凸集。

6.1.3 约束优化测试函数

测试问题 F1：
$$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$
$$subject\ to: \quad x_1 = 2x_2 - 1$$
$$x_1^2 / 4 + x_2^2 - 1 \le 0$$

其最优解 $f^* = 1.3934651$。

测试问题 F2：
$$f_2(x) = (x_1 - 10)^3 + (x_2 - 20)^3$$
$$subject\ to: \quad 100 - (x_1 - 5)^2 - (x_2 - 5)^2 \le 0$$
$$(x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$$
$$13 \le x_1 \le 100, \quad 0 \le x_2 \le 100$$

其最优解 $f^* = -6961.81381$。

测试问题 F3：
$$f_3(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$
$$subject\ to: \quad -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$$
$$-282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$$
$$-196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$$
$$4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$$
$$-10 \le x_i \le 10, \quad i = 1, \cdots, 7$$

其最优解 $f^* = 680.630057$

测试问题 F4： $f_4(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

$$0 \le 85.334407 + 0.0056858T_1 + T_2x_1x_4 - 0.0022053x_3x_5 \le 92$$
$$subject\ to: \quad 90 \le 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110$$
$$20 \le 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25$$
$$78 \le x_1 \le 102, \quad 33 \le x_2 \le 45, \quad 27 \le x_i \le 45, i = 3,4,5$$

其中，$T_1 = x_2x_5$， $T_2 = 0.0006262$ ；最优解 $f^* = -30665.538$。

测试问题 F5：形式与测试问题 F4 相同，只是$T_1 = x_2 x_3$， $T_2 = 0.00026$，最优解未知。

测试问题 F6： $f_6(x) = -10.5x_1 - 7.5x_2 - 3.5x_3 - 2.5x_4 - 1.5x_5 - 10y - 0.5\sum_{i=1}^{5} x_i^2$

$$subject\ to: \begin{array}{c} 6x_1 + 3x_2 + 3x_3 + 2x_4 + x_5 - 6.5 \le 0 \\ 10x_1 + 10x_3 + y \le 20 \\ 0 \le x_i \le 1, \quad i = 1,\ldots,5; \quad y \ge 0 \end{array}$$

最优解 $f^* = -213.0$。

6.1.4 极小极大化测试函数

1. $\min_x F1(x)$ [Xu 2001]

$$F1(x) = \max\{f_i(x)\}, \quad i = 1,2,3$$

$$f_1(x) = x_1^2 + x_2^4$$

$$f_2(x) = (2 - x_1)^2 + (2 - x_2)^2$$

$$f_3(x) = 2\exp(-x_1 + x_2)$$

2. $\min_x F2(x)$ [Xu 2001]

$$F2(x) = \max\{f_i(x)\}, \quad i = 1,2,3$$

$$f_1(x) = x_1^4 + x_2^2$$

$$f_2(x) = (2 - x_1)^2 + (2 - x_2)^2$$

$$f_3(x) = 2\exp(-x_1 + x_2)$$

3. $\min_x F3(x)$ [Xu 2001]

$$F3(x) = \max\{f_i(x)\}, \quad 1 \le i \le m$$

$$f_1(x) = F(x)$$

$$f_i(x) = F(x) - \alpha_i g_i(x), \quad \alpha_i > 0, \quad 2 \le i \le m$$

此处 $F(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1^2 - 21x_3 + 7x_4$

$$g_2(x) = -x_1^2 - x_2^2 - x_3^3 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8$$

$$g_3(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10$$

$$g_4(x) = -x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5$$

$$x_i \in [-5,5]$$

4. $\min\limits_x F4(x)$ [Xu 2001]

$$F4(x) = \max\{f_i(x)\}, \quad 1 \le i \le m$$

$$f_1(x) = F(x)$$

$$f_i(x) = F(x) - \alpha_i g_i(x), \quad \alpha_i > 0, \quad 2 \le i \le m$$

其中 $F(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + 3(x_4 - 11)^2 + x_3^4 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$

$$g_2(x) = -2x_1^2 - 3x_3^4 - x_3 - 4x_4^2 - 5x_5 + 127$$

$$g_3(x) = -7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 + 282$$

$$g_4(x) = -23x_1 - x_2^2 - 6x_6^2 + 8x_7 + 196$$

$$g_5(x) = -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7$$

$$x_i \in [-5,5]$$

5. $\min\max\{|x_1 + 2x_2 - 7|, |2x_1 + x_2 - 5|\}$, $x_i \in [-5,5]$ [Schwefel 1995]

6. $\min\max\{|x_i|\}, 1 \le i \le 10$ [Schwefel 1995]

6.2 设计 PSO 算法的基本原则与步骤

6.2.1 设计 PSO 基本原则

与其他进化算法一样，在设计 PSO 算法时应遵循如下原则：

1、 适用性原则

一个算法的适用性，是指该算法所能适用的问题种类，这取决于算法所需的限制与假设。如果优化问题的性质不同，则相应的具体处理方式也不同。

2、 可靠性原则

一个算法的可靠性，是指算法具有以适当的精度求解大多数问题的能力，即能对大多数问题提供一定精度的解。

与其他众多的进化算法一样，PSO 同样是一种随机的优化算法，因此在求解不同问题时，其结果具有一定的随机性与不确定性。故设计算法实验时，应尽量经过较多样本的检验，以确定算法的可靠性.

3、 收敛性原则

PSO 算法的收敛性是指算法能否以概率 1 收敛到全局最优解,并具有一定的收敛速度和收敛精度。通常评价算法的收敛性能，可通过比较在有限时间代内算法所求解的精度来实现。

4、 稳定性原则

算法的稳定性是指算法对其控制参数及问题数据的敏感程度。一个性能稳定的算法，应该具有以下两种特性：一是对一组固定的控制参数，算法能用于较广泛问题的求解；二是对给定的问题数据，算法的求解结果应不会随控制参数的微小扰动而波动。

5、 生物类比原则.

微粒群算法的设计思想源于对生物群社会行为的模拟，因此生物界中相关的进化理论、策略、方法，均可通过类比的原则，引入到算法中以求提高算法性能。

6.2.2 PSO 算法的设计步骤

不同模型的 PSO 算法，均遵守以下的设计步骤：

1．确定问题的表示方案(编码方案)

和其它的进化算法一样，PSO 算法在求解问题时，首先应先将问题的解从解空间映射到具有某种结构的表示空间，即用特定的码串表示问题的解。根据问题的特征选择适当的编码方法，将会对算法的性能及求解结果产生直接的影响。

PSO 算法的早期研究均集中在数值优化领域中，其标准计算模型适用于具有连续特征的问题函数，因此目前算法大多采用实数向量的编码方案。用 PSO 算法求解具有离散特征的问题对象，正是此领域内的一个研究重点。

2．确定优化问题的评价函数

在求解过程中，借助于适应值来评价解的质量。因此在求解问题时，必须根据问题的具体特征，选取适当的目标函数或费用函数来计算适应度。适应度是唯一能够反映并引导优化过程不断进行的参量。

3．选取控制参数

PSO 算法的控制参数，通常包括微粒群的规模（微粒的数目）、.算法执行的最大代数，惯性系数、认知参数、.社会参数及其他一些辅助控制参数等等。针对不同的算法模型，选取适当的控制参数，直接影响算法的优化性能。

4．设计微粒的飞行模型。

在微粒群算法中，最关键的操作是如何确定微粒的速度。由于微粒是由多维向量来描述的，故相应的微粒的飞行速度也表示为一个多维向量。在飞行过程中，微粒借助于自身的记忆（Lbest）与社会共享信息（Gbest），沿着每一分量方向动态地调整自己的飞行速度与方向。

5．确定算法的终止准则

与其它进化算法一样，PSO 算法中最常用的终止准则是预先设定一个最大的飞行代数，或者是当搜索过程中解的适应度在连续多少代后不再发生明显改进时，终止算法。

6．编程上机运行。

根据所设计的算法结构编程，并进行具体优化问题的求解。通过所获得问题的解的质量，可以验证算法的有效性，准确与可靠性。

6.2.3 PSO 算法的伪码描述

下面是 PSO 算法的伪码描述，其中用到的符号含义如下：

N：微粒群规模　　D：描述微粒的多维向量的维数

P：微粒飞行所经历的最好位置　　G：所有微粒飞行经过的最好位置

Lbest：P 所对应的适应值　　Gbest：G 所对应的适应值

t：迭代次数

Procedure　PSO

begin

t=0

Lbest=Lbest(0)

Gbest=Gbest(0)

while (i<N)

while (d<D)

在[$X_{\min}$，$X_{\max}$]内随机产生 X_{id}

在[$V_{\min}$，$V_{\max}$]内随机产生V_{id}

end while

计算微粒 X_i 的适应度 $fitness_i$

if ($fitness_i$ >Gbest)

令 G= X_i

更新 Lbest, Gbest

end while

t=t+1

while (终止条件未满足时)

while (i<N)

while (d<D)

计算V_{id}

计算 X_{id}

end while

计算微粒 X_i 的适应度 $fitness_i$

if ($fitness_i$ >Gbest)

令 G= X_i

if ($fitness_i$ >Lbest)

令 P= X_i

更新 Lbest, Gbest

end while

end while

end begin

6.3 几种典型的 PSO 模型及其参数选择

自 1995 年微粒群算法问世以来，不同领域的研究人员曾提出各种算法模型，从不同角度对微粒群算法进行了详细的分析、设计。其中倍受推崇的是 Kenney、Eberhart、Yuhui Shi 与 Clerc 等人，他们依据微粒群算法的模拟思想，分别构造了基本微粒群算法模型、含有惯性权重因子的 PSO 模型以及引入收缩因子的 PSO 模型，并通过大量的实验研究，详细分析了模型中不同控制参数的意义与作用，并为其确定了相应的参考取值。这三种算法模型被后来的研究人员视为微粒群算法的经典及研究基础。

下面将会向大家详细介绍这三种模型，同时借助于他们的研究成果，我们共同来探讨微粒群算法的设计及参数选择问题。

6.3.1 简单微粒群算法模型(Simple PSO Model)

Eberhart and Kennedy 定义了最早的微粒群模型，模型如下：

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_1 r_1\left(P_{id}(t) - x_{id}(t)\right) + c_2 r_2\left(P_{gd}(t) - x_{id}(t)\right) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \tag{6.1}$$

此模型中含有两个控制参数 c_1， c_2，可视为加速度常量。其中 c_1 反映了微粒飞行过程中所记忆的最好位置(P)对微粒飞行速度的影响，又被称为“认知系数”；c_2 则反映了整个微粒群所记忆的最好位置(Gbest)对微粒飞行速度的影响，又称为“社会学习系数”。

为确定 c_1， c_2 对算法性能影响，1997 年 Kennedy 在其所发表的“The Particle Swarm：social adoptions of Knowledge”一文中，通过下述系列实验，详细讨论了两个参数对算法性能的作用[Kennedy,1997]。

实验一：令 c_2 =0, (认知模型, the cognition-only model)

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_1 r_1\left(P_{id}(t) - x_{id}(t)\right) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \tag{6.2}$$

实验二：令 $c_1 = 0$,(社会模型, the social-only model)

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_2 r_2\left(P_{gd}(t) - x_{id}(t)\right) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \tag{6.3}$$

实验三：令 $c_1 = c_2$ =C(完全模型, the full model)

实验四：自私模型（the selfness model）

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_2 r_2\left(P_{gd}(t) - x_{id}(t)\right), \quad g \neq i \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \tag{6.4}$$

自私模型也属于一种社会模型，唯一的区别在于每个微粒在获取 Gbest 时，只考虑其他微粒的相关信息，而不包括自己。

Kennedy 将 PSO 算法用于 XOR 神经网络权值的优化，比较了不同模型在取不同参数时的优

化结果，发现模型 1 的求解性能最差，模型 2 与 4 的求解性能要优于模型 3。同时 Kennedy 还发现，算法在执行的过程中，微粒的飞行速度应当受到适当的限制，一方面保证微粒能够飞越局部极值，具有一定的全局探测能力；另一方面又能够使得微粒以一定的速度步长逼近全局最优解。通常可在计算模型中引入一个适当的阈值 Vmax。

随后 Kennedy 在文献[Kennedy 1998]中进一步研究了两种参数的具体取值，经过大量的实验结果对比，指出c_1与c_2之和最好接近 4.0，通常$c_1 \approx c_2 = 2.05$。

6.3.2 引入惯性权重系数ω的 PSO

从基本微粒群算法模型可以看出，微粒的飞行速度相当于搜索步长，其大小直接影响着算法的全局收敛性。当微粒的飞行速度过大时，能够保证各微粒以较快的速度飞向全局最优解所在的区域。但是当逼近最优解时，由于微粒的飞行速度缺乏有效的控制与约束，则将很容易飞越最优解，转而去探索其它区域，从而使算法很难收敛于全局最优解。这一现象，同时也说明了算法在速度缺乏有效的控制策略时，不具备较强的局部搜索能力（或精细搜索能力）。

为了有效地控制微粒的飞行速度，使得算法能够达到全局探测与局部开采功能间的有效平衡，单依靠施加 Vmax 是不够的。文献[Shi and Eberhart，1998]在算法模型中引入了惯性权重系数ω，以实现对微粒飞行速度的有效控制与调整。这一思想源于模拟退火算法，ω类似于其中的温度控制参数。具体的计算模型如下：

$$\begin{aligned} v_{id}(t+1) &= \omega v_{id}(t) + c_1 r_1\left(P_{id}(t) - x_{id}(t)\right) + c_2 r_2\left(P_{gd}(t) - x_{id}(t)\right) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \tag{6.5}$$

从上式易知，ω越大，则微粒的飞行速度越大，微粒将以较大的步长进行全局探测；ω越小，则微粒的速度步长越小，微粒将趋向于进行精细的局部搜索。Shi and Eberhart 等人经过实验发现，当$\omega \in [0.9, 1.2]$时，算法具有较理想的搜索性能。另外，在搜索过程中可以对ω进行动态调整：在算法开始时，可以给ω赋予一较大正值，随着搜索的进行，可以线性地使ω逐渐减小，这样可以保证在算法开始时，各微粒能够以较大的速度步长在全局范围内探测到较好的种子；而在搜索后期，较小的ω值则保证微粒能够在极点周围做精细地搜索，从而使算法有较大的机率以一定精度收敛于全局最优解。为了确定惯性系数ω对算法性能的影响，Shi and Eberhart 设计了如下一系列实验[Shi and Eberhart,1998,1999]

实验一：用引入惯性系数ω的 PSO 模型求解优化函数(F_1, F_2, F_3, F_4)，观察微粒群规模对算法的影响。控制参数取值如下：$c_1 = c_2 = 2$，$V_{\max} = X_{\max}$，微粒群规模 N 分别为 20、40、80、160，ω从 0.9 减小至 0. 4，部分实验结果见下表：

表 6.1　算法采用不同微粒群规模时的求解结果

N	Dimension	Generation	$MBF\ \ F_1$	$MBF\ \ F_2$	$MBF\ \ F_3$	$MBF\ \ F_4$
20	10	1000	0.0000	96.1715	5.5572	0.0919
	20	1500	0.0000	214.6764	22.8892	0.0303
	30	2000	0.0000	316.4468	47.2941	0.0182

40	10	1000	0.0000	70.2139	3.5623	0.0862
	20	1500	0.0000	180.9671	16.3504	0.0286
	30	2000	0.0000	299.7061	38.5250	0.0127
80	10	1000	0.0000	36.2945	2.53579	0.0760
	20	1500	0.0000	87.2802	13.4263	0.0288
	30	2000	0.0000	205.5596	29.3063	0.0128
160	10	1000	0.0000	24.4477	1.4943	0.0628
	20	1500	0.0000	72.8190	10.3696	0.0300
	30	2000	0.0000	131.5866	24.0864	0.0127

注：MFB=Mean Best Fitness

根据系列实验数据，Shi and Eberhart 得出下述结论：在算法执行过程中，将W 随时间逐渐减小，这种方法在一定程度上会导致搜索后期收敛速度的降低，从而影响算法的全局收敛性能，但与采用固定ω 的 PSO 算法与 EP 算法相比较，其搜索结果仍然有很大地提高与改善。同时，他们发现群体规模的大小对 PSO 算法的性能并没有太大的影响。

实验二：比较ω 与$V_{\max}$ 两个参数对算法性能的影响，分析两者在算法中不同的地位。实验中利用 PSO 算法去求解 Schaffer 函数，Shi 等人详细观察了取不同的$V_{\max}$ 、不同的ω 值时算法的运行结果，并统计了算法在 30 次运行过程中收敛时所需的叠代次数，部分实验数据如下：

表 6.2 算法取不同ω 与 $V_{\max}$ 收敛时的平均代数

	Weights								
$V_{\max}$	1.4	1.2	1.1	1	0.9	0.8	0.7	0.6	0
3	2387	1840	1758	1653	738	438	402		663
4	---	2215	2128	1967	946	439	471	421	652
5	---	---	---	2456	1090	366	503	535	1133
10	---	---	---	658	2027	460	789	490	895
$X_{\max}$	---	---	---	---	---	974	879	927	933

注：虚线表格表示没有统计数据

表 6.3　采用随时间递减的ω 值 30 次运行中的收敛代数（$V_{\max} = X_{\max}$）

Index	Iterations	Index	Iterations	Index	Iterations
1	423	11	429	21	407
2	231	12	398	22	309
3	317	13	427	23	305
4	373	14	362	24	421
5	321	15	338	25	394
6	226	16	510	26	243
7	250	17	373	27	337
8	241	18	302	28	268

9	293	19	402	29	378
10	284	20	461	30	359
Average					344

通过对实验数据的分析，Shi 与 Eberhart 得到下述结论：

①$V_{\max}$ 间接地影响算法的全局搜索能力。

当$V_{\max} \le 2$时　ω 取值最好选择接近于 1。

当$V_{\max} \ge 3$时　ω=0.8 是最佳的选择。当$\omega \in (0.9,1.2)$时，一般可达到比较理想的结果。

当$V_{\max}$ 取值过小时，则无论 ω 取值如何，算法总是趋向于局部搜索，而缺乏全局探测能力；如果$V_{\max} = X_{\max}$ 足够大，则算法的性能主要取决于 ω。

②算法可仅选择 ω 作为控制参数。执行过程中可使 ω 从一个较大的初始值逐渐减小（1.4→0）同样可提高算法的搜索性能。但是在搜索后期，较小的 ω 值一定程度上限制了算法的全局搜索能力。

然而 Shi and Eberhart 的结论受到了质疑，Ozcan and Mohan 在文献[Ozcan and Mohan 1998]一文中分析了 PSO 在多维搜索空间的能力，速度步长及 ω 系数对算法的影响。他们所得结论与 Shi 有所矛盾，他们认为 ω 并不能显著提高算法的性能，当 ω 接近于 1 时，实验结果最佳，说明 ω 对算法的有效性可能被算法本身固有的随机性而克服掉。

6.3.3　引入收缩因子的 PSO 模型

出于同样的目的，为了有效地控制微粒的飞行速度，使算法达到全局探测与局部开采两者间的有效平衡，Clerc 构造了引入收缩因子的 PSO 模型[Clerc 1999]。模型如下：

$$\begin{aligned} v_{id}(t+1) &= K(v_{id}(t) + c_1 r_1 (P_{id}(t) - x_{id}(t)) + c_2 r_2 (P_{gd}(t) - x_{id}(t))) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \tag{6.6}$$

其中，K 称为收缩因子，$K = \dfrac{2}{\left|2 - C - \sqrt{c_2 - 4C}\right|}$，$C = c_1 + c_2$且$C > 4$

此模型中 K 的作用类似于参数$V_{\max}$ 的作用，用来控制与约束微粒的飞行速度。但实验结果表明，K 比$V_{\max}$ 更能有效地控制微粒速度的振动。Eberhart and Shi 在文献[Eberhart and Shi 1999]中，详细分析比较了 ω 与 K 两种参数对 PSO 算法性能的影响，他们谦虚地承认 Clerc 所提出的收缩因子 K 比起惯性权重系数 ω，更能有效地控制与约束微粒的飞行速度，同时增强了算法的局部搜索能力。

2001 年，Carlisle 与 Dozier 综合了 PSO 参数选择问题上的各种研究成果，为 PSO 提出了一组较完善、合理的参数，称之为经典 PSO 参数集[Carlisle and Dozier 2001]。

6.3.4　经典 PSO 算法（The Canonical PSO， CPSO）

综合上述的种种研究可知， PSO 算法中通常有如下几种重要的控制参数：

1、微粒群规模（Population Size）：指微粒群中所包含的微粒个数。

2、认知学习系数 c_1（Cognitive learning rate）

3、社会学习系数 c_2（Social learning rate）

4、控制微粒飞行速度的 $V_{\max}$

5、惯性权重系数 ω（inertia weight factor）

6、收缩因子 K（Constriction factor）

Carlisle 与 Doziert 通过系列的实验研究，详细分析了上述不同控制参数对算法性能的影响，在此基础上提出了下述的经典 PSO 算法模型。

$$v_{id} = \begin{cases} K(v_{id} + c_1 r_1 (P_{id} - x_{id}) + c_2 r_2 (P_{gd} - x_{id})), & X_{\min} < x_{id} < X_{\max} \\ 0 & \text{其它} \end{cases}$$

$$x_{id} = \begin{cases} x_{id} + v_{id}, & X_{\min} < x_{id} < X_{\max} \\ X_{\max}, & (x_{id} + v_{id}) > X_{\max} \\ X_{\min}, & X_{\min} < (x_{id} + v_{id}) \end{cases} \tag{6.7}$$

其中 K 为收缩因子，$K = \dfrac{2}{\left|2 - C - \sqrt{c_2 - 4C}\right|}$ 各控制参数的最佳取值为：

$c_1 = 2.8,\quad c_2 = 1.3,\quad C = c_1 + c_2$，微粒群规模 N=30。这些参数的组合被称为经典参数集。