

# 微粒群算法的统一模型及分析\*

曾建潮 崔志华

太原科技大学 系统仿真与计算机应用研究所, 030024

E\_mail: zengjianchao@263.net

**摘 要:** 通过分析已有的几种微粒群算法, 提出了一种统一模型, 并通过线性控制理论分析了其收敛性能, 同时利用李亚普诺夫函数给出了模型收敛速度的上界估计。为了进一步提高算法效率, 提出了两种增强全局搜索性能的参数自适应算法: 单群体参数自适应微粒群算法及双群体参数自适应微粒群算法。其中单群体参数自适应微粒群算法在进化初期使用算法发散的参数设置, 从而能更大程度的提高算法全局收敛能力。双群体参数自适应微粒群算法使用两个种群, 一个执行全局搜索, 另一个执行局部搜索, 通过信息交流以提高算法性能。仿真实例证明了算法的有效性。

**关键词:** 统一模型 收敛性 自适应 微粒群算法

## A New Unified Model of Particle Swarm Optimization and Its Theoretical Analysis

Zeng Jianchao Cui Zhihua

Division of System Simulation And Computer Application, Taiyuan University of Science and Technology, 030024

**Abstract:** Through mechanism analysis of several modified particle swarm optimizations (PSO), a new uniform model of PSO is described, and the convergence is analysed with linear control theory. Through Layopov function, the upper estimate of the model convergence speed is given. To improve the calculation efficiency, two enhanced global search capability self-adaptive PSOs, one-population self-adaptive PSO and two-population self-adaptive PSO, are proposed. One-population self-adaptive PSO using the diverse coefficients in the first evolutionary strategy. Two-population self-adaptive PSO using two different populations, one owns global search capability, the other owns local search, and through exchanging information to improve algorithm efficiency. The simulation results show the correctness and efficiency of the presented methods.

**Keywords:** Unified Model Convergence Self-adaptive Particle Swarm Optimization

### 一、引言。

群体智能算法 (Swarm Intelligence Algorithm) 的研究开始于 20 世纪 90 年代初, 其基本思想是模拟自然界生物的群体行为来构造随机优化算法。如蚁群算法<sup>[1]</sup>和微粒群算法 (Particle Swarm Optimizer, PSO)。微粒群算法是在 1995 年由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 共同提出的<sup>[2]</sup>, 其基本思想是受他们早期对鸟类群体行为研究结果的启发, 并利用了生物学家 Frank Heppner 的生物群体模型。

为了提高基本微粒群算法的计算效率, Yuhui Shi<sup>[3]</sup>引入了带有惯性权重的改进微粒群算法, 在 Clerc<sup>[4]</sup>的研究中, 提出了收缩因子的概念。Angeline<sup>[5]</sup>提出了杂交微粒群算法, 微粒群中的微粒被赋予一个杂交概率, 这个杂交概率是用户确定的, 与微粒的适应值无关。在每次迭代中, 依据杂交概率选取指定数量的微粒放入一个池中。池中的微粒随机地两两杂交, 产生相同数目的子代, 并用子代微粒取代父代微粒, 以保持种群的微粒数目不变。P.N. Suganthan<sup>[6]</sup>于 1999 年提出一种基于邻域思想的微粒群算法, 其基本思想是在算法开始

\* 本文受教育部科学技术研究重点项目 (编号: 204018) 资助

阶段, 每个个体的邻域为其自身, 随着进化代数的增长, 其邻域范围也在不断增大直至整个种群。为了避免微粒群算法所存在的过早收敛问题, J.Riget<sup>[7]</sup>提出了一种保证种群多样性的微粒群算法 (Attractive and Repulsive Particle Swarm Optimizer, 简称ARPSO)。该算法引入“吸引”(attractive)和“扩散”(repulsive)两个算子, 动态地调整“勘探”与“开发”比例, 从而能更好的提高算法效率。J.Kennedy 和Eberhart<sup>[8]</sup>在 1997 年将基本微粒群算法应用于二进制编码, 并作了大量的数值研究。

有关PSO算法的全局收敛性<sup>[9]</sup>研究, 大多针对典型优化问题进行仿真实验研究, 给出全局最优性的实验结果。F.Solis和R.Wets<sup>[10]</sup>对随机优化算法提出了其全局收敛须满足的条件, Frans Van Den Bergh<sup>[11]</sup>利用该条件对基本PSO算法和保证收敛的PSO算法 (GCP SO) 的全局收敛性和局部收敛性进行了研究, 指出基本PSO算法不能保证全局或局部收敛, 而GCP SO则属于局部收敛。我们在对基本PSO算法进行分析的基础上, 提出了一种速度“无记忆”的随机PSO算法<sup>[12]</sup>, 该算法通过修改速度进化方程, 同时, 给出了几种随机微粒的产生方式, 并证明了该算法能够保证以概率 1 全局收敛。

为了对已有各种 PSO 进化方程进行分析, 本文提出了一种统一模型, 并对模型的进化行为进行了理论分析, 同时给出了保证 PSO 算法具有全局收敛性的参数自适应方案。

## 二、 PSO 算法的统一模型。

考察下列微分方程组

$$\begin{cases} \frac{dV(t)}{dt} = \chi[(w - \frac{1}{\chi})V(t) + c_1 r_1 (P_i - X(t)) + c_2 r_2 (P_g - X(t))] \\ \frac{dX(t)}{dt} = V(t+1) \end{cases} \quad (1)$$

- (1) 当  $w = 1, \chi = 1$  时, 采用步长为 1 的 Euler 法即可得到基本 PSO 进化方程;
- (2) 当  $w \neq 1, \chi = 1$  时, 采用步长为 1 的 Euler 法即可得到标准 PSO 进化方程;
- (3) 当  $w = 1, \chi \neq 1$  时, 采用步长为 1 的 Euler 法即可得到带收缩因子的 PSO 进化方程;
- (4) 当  $w = 0, \chi = 1$  时, 采用步长为 1 的 Euler 法即可得到随机 PSO 进化方程。

也就是说, 当参数  $\chi, w$  取不同值时, 方程 (1) 代表了不同的 PSO 进化方程。因此, 方程 (1) 可以作为各种 PSO 进化方程的统一描述模型。为了后面分析的方便, 定义:

$$\varphi_1 = c_1 r_1, \quad \varphi_2 = c_2 r_2, \quad \varphi_{12} = \varphi_1 + \varphi_2, \quad \varphi_p = \varphi_1 P_i + \varphi_2 P_g$$

则上述微分方程组可写为:

$$\begin{cases} \frac{dv(t)}{dt} = \chi[(w - \frac{1}{\chi})v(t) - \varphi_{12}x(t) + \varphi_p] \\ \frac{dx(t)}{dt} = v(t+1) \end{cases} \quad (2)$$

### 三、 基于统一描述模型的 PSO 进化行为分析。

将  $v(t+1)$  作一阶近似, 即  $v(t+1) = v(t) + \frac{dv(t)}{dt}$ , 代入 (2) 式并整理, 得到

$$\begin{cases} \frac{dv(t)}{dt} = (w\chi - 1)v(t) - \chi\varphi_{12}x(t) + \chi\varphi_p \\ \frac{dx(t)}{dt} = w\chi v(t) - \chi\varphi_{12}x(t) + \chi\varphi_p \end{cases} \quad (3)$$

定义

$$y(t) = \begin{bmatrix} v(t) \\ x(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} p_i \\ p_g \end{bmatrix}, \quad A = \begin{bmatrix} \chi w - 1 & -\chi\varphi_{12} \\ \chi w & -\chi\varphi_{12} \end{bmatrix}, \quad B = \begin{bmatrix} \chi\varphi_p & \chi\varphi_p \\ \chi\varphi_p & \chi\varphi_p \end{bmatrix}$$

则有

$$\dot{y}(t) = Ay(t) + Bu(t) \quad (4)$$

方程 (4) 的解为:

$$y(t) = e^{A(t-t_0)} y(t_0) + \int_{t_0}^t e^{A(t-\tau)} Bu(\tau) d\tau \quad (5)$$

从上式可以看出, 当矩阵  $A$  的特征根具有负的实部时, 方程 (5) 收敛。其特征方程为

$$|\lambda I - A| = \lambda^2 + (1 - \chi w + \chi\varphi_{12})\lambda + \chi\varphi_{12} = 0 \quad (6)$$

其特征根为

$$\lambda_{1,2} = \frac{\chi w - \chi\varphi_{1,2} - 1 \pm \sqrt{(1 - \chi w + \chi\varphi_{1,2})^2 - 4\chi\varphi_{1,2}}}{2} \quad (7)$$

只要  $\chi w - \chi\varphi_{1,2} - 1 < 0$ , 就可以保证矩阵  $A$  的特征根为负的实部, 也就是说, 只要

$\chi w - \chi\varphi_{1,2} - 1 < 0$ , 由 (3) 描述的 PSO 进化方程收敛。而

$$\lim_{t \rightarrow +\infty} y(t) = \lim_{t \rightarrow +\infty} e^{A(t-t_0)} y(t_0) + \int_{t_0}^t e^{A(t-\tau)} Bu(\tau) d\tau = \begin{bmatrix} 0 \\ \frac{\varphi_p}{\varphi_{1,2}} \end{bmatrix}$$

即:

$$\lim_{t \rightarrow +\infty} v(t) = 0, \quad \lim_{t \rightarrow +\infty} x(t) = \frac{\varphi_p}{\varphi_{1,2}}$$

也就是

$$\lim_{t \rightarrow +\infty} (\varphi_1 + \varphi_2)x(t) = \varphi_1 p_i + \varphi_2 p_g$$

由于  $\varphi_1, \varphi_2$  为随机变量, 要使上式成立, 当且仅当  $\lim_{t \rightarrow +\infty} x(t) = p_i = p_g$ 。

从上述分析可以看出, 当  $\chi w - \chi \varphi_{1,2} - 1 < 0$  时, 所有微粒将收敛于全局最优位置  $p_g$ 。也就是说, 当  $p_g$  固定不变时,  $x(t)$  收敛于  $p_g$ 。因此, 在 PSO 算法的进化过程中, 只要能探索到全局最优位置  $x^*$ , 即  $p_g = x^*$ , 就能保证所有微粒将最终收敛于  $p_g = x^*$ 。

另外, 由方程 (4) 及  $u(t) = \begin{bmatrix} p_i \\ p_g \end{bmatrix}$  知, 微粒群算法实际上是在阶跃信号输入下的线性系统, 且其输入阶跃信号的幅值随进化而变化。

下面给出方程 (4) 收敛的上界估计。定义李亚普诺夫函数

$$V(y) = y^T P y \quad (8)$$

$$\dot{V}(y) = -y^T Q y \quad (9)$$

其中,  $P$  为正定阵,  $Q$  为正定对称阵, 且满足下列李亚普诺夫方程

$$A^T P + P A = -Q \quad (10)$$

系统的收敛性能可用

$$\eta = -\frac{\dot{V}(y)}{V(y)} \quad (11)$$

来表征。显然,  $V(x)$  愈小且  $\dot{V}(x)$  的绝对值愈大则  $\eta$  愈大, 从而系统收敛愈快; 反之, 则  $\eta$  愈小, 相应的系统收敛愈慢。

进一步, 对 (11) 式由  $t=0$  到  $t$  进行积分, 可得到

$$-\int_0^t \eta dt = \int_0^t \frac{\dot{V}(y)}{V(y)} dt = \ln \frac{V(y)}{V(y_0)} \quad (12)$$

从而,

$$V(y) = V(y_0) e^{-\int_0^t \eta dt} \quad (13)$$

一般的说, 直接由 (13) 式是难以进行估计的, 取

$$\eta_{\min} = \min_y \left\{ -\frac{\dot{V}(y)}{V(y)} \right\} = \text{常数} \quad (14)$$

并将其代入 (13) 式, 可以得到

$$V(y) \leq V(y_0) e^{-\int_0^t \eta_{\min} dt} = V(y_0) e^{-\eta_{\min} t} \quad (15)$$

这表明, 一旦确定出  $\eta_{\min}$ , 就可以定出  $V(x)$  随时间的收敛上界。由 (8)、(9) 两式, 有

$$\eta_{\min} = \min_y \left\{ -\frac{\dot{V}(x)}{V(x)} \right\} = \min_y \left\{ \frac{y^T Q y}{y^T P y} \right\} = \min_y \{ y^T Q y, y^T P y = 1 \} \quad (16)$$

对于线性定常系统:

$$\eta_{\min} = \lambda_{\min}(QP^{-1}) \quad (17)$$

其中  $\lambda_{\min}(\cdot)$  表示  $(\cdot)$  的最小特征值。

下面给出几种典型 PSO 进化方程的收敛条件。由 (7) 式知, PSO 进化方程渐进收敛的条件为:

$$\begin{cases} \chi w - \chi \varphi_{1,2} - 1 < 0 \\ (1 - \chi w + \chi \varphi_{1,2})^2 - 4\chi \varphi_{1,2} \geq 0 \end{cases} \quad (18)$$

(1) 当  $\chi = w = 1$  时, 为基本 PSO 进化方程, 渐进收敛条件为

$$\begin{cases} \varphi_{1,2} \geq 0 \\ \varphi_{1,2}^2 \geq 4\varphi_{1,2} \end{cases} \quad (19)$$

即: 当  $\varphi_{1,2} \geq 4$  时, 基本 PSO 进化方程渐进收敛, 而当  $0 \leq \varphi_{1,2} < 4$  时, 系统有两个具有负实部的复根, 基本 PSO 进化方程震荡收敛。

(2) 当  $\chi = 1$  时, 为标准 PSO 进化方程, 渐进收敛条件为

$$\begin{cases} w - \varphi_{1,2} - 1 < 0 \\ (1 - w + \varphi_{1,2})^2 - 4\varphi_{1,2} \geq 0 \end{cases}$$

从图 1 中可以看出, 下半部分为渐进收敛区域, 上半部分为震荡收敛区域。

(3) 当  $w=1$ ,  $\chi = \frac{2}{|2 - \varphi_{1,2} - \sqrt{\varphi_{1,2}^2 - 4\varphi_{1,2}}|}$  时, 为带收缩因子的 PSO 进化方程, 当

$\varphi_{1,2} > 4$  时, 有

$$\begin{cases} \chi - \chi \varphi_{1,2} - 1 < 0 \\ (1 - \chi + \chi \varphi_{1,2})^2 - 4\chi \varphi_{1,2} \geq 0 \end{cases}$$

因而系统渐进收敛。

#### 四、 增强全局搜索能力的参数自适应算法。

从上述的 PSO 进化行为分析可以看出, PSO 的收敛性是很容易保证的。也就是说, PSO 算法具有良好的局部搜索能力。为了使 PSO 算法具有良好的全局搜索能力, 我们提出了两种改进的微粒群算法: 单群体和双群体参数自适应微粒群算法。可以设想在单群体参数自适应微粒群算法的初始进化阶段, 让算法处于发散过程, 即可增加 PSO 算法的搜索区域, 增加群体的多样性; 而在 PSO 算法的进化中期阶段, 让算法处于震荡收敛过程, 即可保证在收敛的前提下增加搜索区域; 最后在 PSO 算法的进化后期, 为了加快收敛速度, 让算法处

于渐进收敛过程。对于双群体参数自适应微粒群算法而言,使用两个不同的种群,其中一个专门负责全局搜索,通过选择保证微粒群算法发散的参数设置,不断开拓新的搜索区域,从而可有效避免算法的过早收敛现象。另一个种群先使用保证微粒群算法渐进收敛的参数设置,到一定程度后再使用保证微粒群算法收敛的参数设置,并利用前一个种群的信息以提高算法效率。

依据上述思想,给出标准 PSO 算法的参数自适应方法。由(20)式知,当  $w > 1 + \varphi_{1,2}$  时, PSO 算法不收敛;当  $1 + \varphi_{1,2} - 2\sqrt{\varphi_{1,2}} < w < 1 + \varphi_{1,2}$  时, PSO 算法震荡收敛;而当  $w < 1 + \varphi_{1,2} - 2\sqrt{\varphi_{1,2}}$  时, PSO 算法渐进收敛。

为了引入新的改进微粒群算法,我们首先将微粒的进化方程修改为<sup>[10][11]</sup>

$$V_j(t+1) = wV_j(t) + \varphi(P_Q - X_j(t))$$

$$X_j(t+1) = X_j(t) + V_j(t+1)$$

$$P_Q = \frac{\varphi_1 P_j + \varphi_2 P_g}{\varphi}$$

且参数  $W$  由  $\varphi$  决定。

首先给出参数  $\varphi$  的计算方法。由于  $r_1, r_2 \in [0,1]$  为均匀分布的随机数,  $\varphi_1 = c_1 r_1$  为  $[0, c_1]$  上的均匀分布,  $\varphi_2 = c_2 r_2$  为  $[0, c_2]$  上的均匀分布,从而  $\varphi = \varphi_1 + \varphi_2$  为  $[0, c_1 + c_2]$  上的随机变量。当然,由于  $c_1, c_2$  的选择,仅仅有一些试验结果,因而  $\varphi$  可用变量  $K \cdot \text{random}$ , 其中  $K$  为一给定的数,用以限制  $\varphi$  的变化范围,而  $\text{random}$  则是介于 0, 1 间的随机数。

其次给出惯性系数  $W$  的计算方法。即在单群体参数自适应微粒群算法中,惯性系数  $W$  可按照下式进行计算:

$$W = \begin{cases} \varphi_{1,2} + 1.1, & \text{if } (Gene < 100) \\ \varphi_{1,2} - \sqrt{\varphi_{1,2}} + 1.0, & \text{if } (Gene < 200) \\ \varphi_{1,2} - 2\sqrt{\varphi_{1,2}} + 0.9, & \text{otherwise} \end{cases}$$

而对于双群体参数自适应微粒群算法而言,其进行全局搜索的微粒群进化方程中的惯性系数  $W$  的计算公式为:

$$W = \varphi_{1,2} + 1.1$$

进行局部搜索的微粒群进化方程中的惯性系数  $W$  的计算公式为

$$W = \begin{cases} \varphi_{1,2} - \sqrt{\varphi_{1,2}} + 1.0, & \text{if } (Gene < 200) \\ \varphi_{1,2} - 2\sqrt{\varphi_{1,2}} + 0.9, & \text{otherwise} \end{cases}$$

此外,在双群体参数自适应微粒群算法中还存在着信息交流的过程,该信息交流采用 Metropolis 准则进行,设个体  $X_k$  为全局搜索微粒群中当前位置最优的个体,个体  $Y_j$  为局部

搜索微粒群中随机选择的某一个体，当  $f(X_k) < f(Y_j)$  或  $e^{\frac{f(Y_j)-f(X_k)}{T}} > random$  时，用个体  $X_k$  的当前位置替换个体  $Y_j$  的当前位置，其中， $T$  为 Metropolis 准则的参数， $random$  为一介于 0, 1 之间的随机数。

## 五、 实例仿真。

为了验证本文算法与带有惯性系数的微粒群算法性能，我们采用如下两个测试函数进行测试。

$$f_1(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)], \quad x_j \in [-2, 2]$$

$$f_2(X) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5, \quad x_j \in [-100, 100]$$

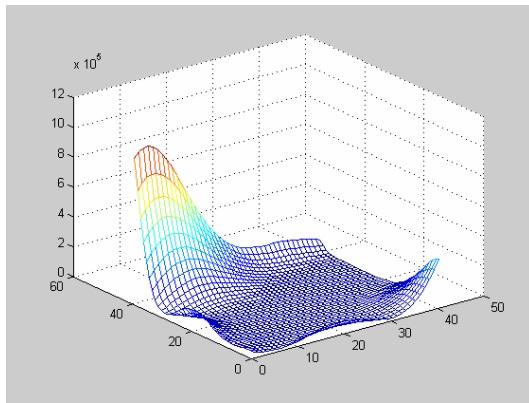
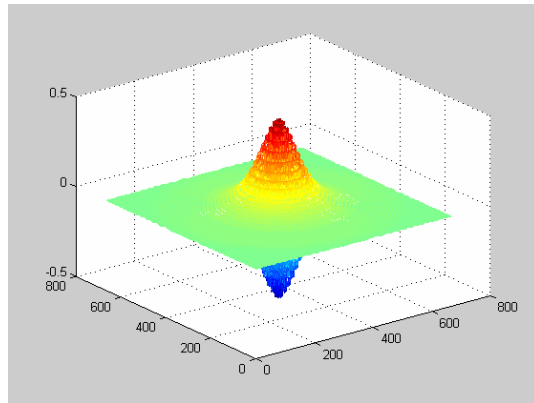
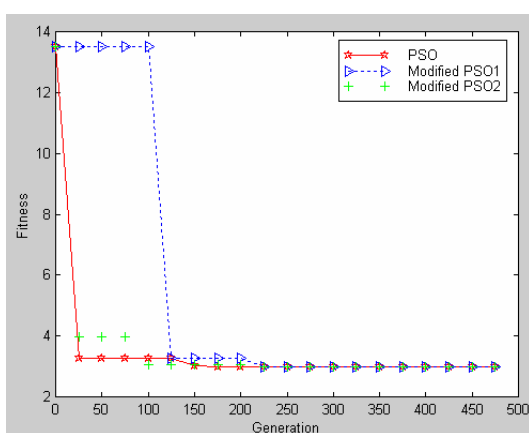
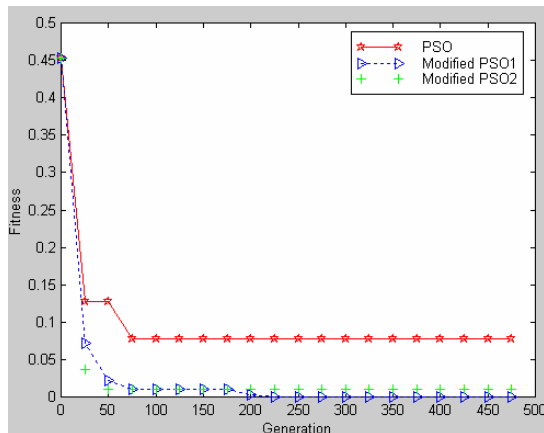
函数  $f_1(X)$  称为 Goldstein-Price 函数，其最优状态和最优值为  $f_1(0, -1) = 3$ 。 $f_2(X)$

是由 J. D. Schaffer 等提出的。它的全局极小点是 (0, 0)；而在距全局极小点约 3.14 范围内的隆起有无限多全局极大点，取值为 -0.990283，因此很容易陷入局部极小点。由于该函数的强烈震荡性质以及它的全局最优点被局部最优点所包围的特性使得一般算法很难找到它的全局最优解。

在基本 PSO 算法中， $W$  从 1.0 到 0.4 随进化而线性减少， $c_1 = c_2 = 1.8$ 。参数  $K$  取 2.0， $T$  的初值为 1.0，其进化公式为  $T = T * 0.9$ 。分别对上述问题进行了 50 次仿真计算，群体规模为 20，最大进化代数为 500，其计算结果如表 1 所示。

表 1、函数  $f_1(X)$ 、 $f_2(X)$  性能比较

函数	算法	误差	平均收敛代数	平均收敛概率
F1	PSO	0.0000001	253.42	100
F1	MPSO1	0.0000001	218.16	100
F1	MPSO2	0.0000001	218.50	100
F2	PSO	0.01	73.904762	42
F2	MPSO1	0.01	103.12	100
F2	MPSO2	0.01	91.806452	62

图 1、函数  $f_1(X)$  的图像图 2、函数  $f_2(X)$  的图像图 3、函数  $f_1(X)$  的性能比较图 4、函数  $f_2(X)$  的性能比较

本文通过分析已有的一些微粒群算法结论,提出一种涵盖标准微粒群算法、基本微粒群算法、带有收缩因子的微粒群算法的统一描述模型,并从理论上讨论了其收敛性,给出了惯性因子  $W$  与参数  $\varphi_{1,2}$  的比例关系。根据这一关系,提出了两种不同的改进微粒群算法。仿真试验表明了算法的有效性。

## 参考文献

- 1、Colomi,A., Dorigo,M.,Maniezzo V., Distributed optimization by ant colonies, In Proc. First Europ. Conference on Artificial Life, Cambridge, MA:MIT Press,1991,134-142.
- 2、J.Kennedy and R.C.Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, 1942-1948, Perth, Australia, 1995, IEEE Service Center, Piscataway, NJ.
- 3、Y.Shi and R.C.Eberhart, A Modified Particle Swarm Optimizer, IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, May 1998.
- 4、M.Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, In Proc. Of the Congress on Evolutionary Computation, 1951-1957, Washington DC, USA, 1999.
- 5、P.J.Angeline, Using Selection to Improve Particle Swarm Optimization, in Proceedings of IJCNN'99, 84-89, Washington D.C, USA, July 1999.
- 6、P.N.Suganthan, Particle Swarm Optimizer with Neighborhood Operator, In Proceedings of the



Congress on Evolutionary Computation,1958-1961, Washington D.C, USA, July 1999, IEEE Service Center, Piscataway, NJ.

7、Riget, J.Krink, T., and Vesterstroem, J. S., Particle Swarm Optimisation with Spatial Particle Extension. 5-12-2002. Honolulu, Hawaii USA. IEEE Congress on Evolutionary Computation, 2002.

8、J.Kennedy and Eberhart R., A Discrete Binary Version of the Particle Swarm Algorithm, IEEE International Conference on Computational Cybernetics and Simulation, 1997:4104-4108.

9、M.Clerc and J.Kennedy, The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space,IEEE Transactiona on Evolutionary Computation, Vol.6,58-73,2002.

10、F.Solis, R.Wets, Minimization by Random Search Techniques[J], Mathematics of Operations Research,6:19-30,1981.

11、F.Van den Bergh, An Analysis of Particle Swarm Optimizers, Ph.D thesis, University of Pretoria, Nov. 2001.

12、曾建潮, 崔志华, 一种保证全局收敛的 PSO 算法, 计算机研究与发展, 已录用。

13、崔志华, 曾建潮, 一种动态调整的改进微粒群算法, 系统工程学报, 已录用。

14、崔志华, 曾建潮, 蔡星娟, 一种保证全局收敛的动态双层微粒群算法, Fifth World Congress on Intelligent Control and Automation, Hangzhou,2184-2188。