

文章编号: 1001-0920(2003) 02-0129-06

## 微粒群算法综述

谢晓锋, 张文俊, 杨之廉

(清华大学 微电子学研究所, 北京 100084)

**摘要:** 讨论微粒群算法的开发与应用。首先回顾从 1995 年以来的开发过程, 然后根据一些已有的测试结果对其参数设置进行系统地分析, 并讨论一些非标准的改进手段, 如簇分解选择方法、邻域算子、无希望/重新希望方法等。介绍了一些常用的测试函数, 以及与其他演化算法的比较。最后讨论了一些已经开发和在将来有希望的领域中的应用。

**关键词:** 微粒群算法; 演化计算; 人工生命

**中图分类号:** TP301.6      **文献标识码:** A

## Overview of particle swarm optimization

XIE Xiao-feng, ZHANG Wen-jun, YANG Zhi-lian

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

**Abstract:** The developments and applications related to particle swarm optimization (PSO) are discussed. Firstly, developments in the particle swarm optimization since 1995 are reviewed. Then parameter settings are analyzed systematically according to some existed testing results. Some improvement methods, such as cluster analysis, selection, neighborhood operator, no-hope/re-hope method, etc., are discussed. Some common test functions and the comparisons between PSO and others evolutionary algorithms are introduced. Finally, applications, both in the developed areas and the promising future application areas, are reviewed.

**Key words:** Particle swarm optimization; Evolutionary computation; Artificial life

### 1 引言

微粒群算法(PSO)是由 Kennedy 和 Eberhart 等<sup>[1, 2]</sup>于 1995 年开发的一种演化计算技术, 来源于对一个简化社会模型的模拟。其中“群(swarm)”来源于微粒群符合 Millonas<sup>[3]</sup>在开发应用于人工生命(artificial life)的模型时所提出的群体智能的 5 个基本原则。而“微粒(particle)”则是一个折衷的选择, 因为既需要将群体中的成员描述为没有质量、没有体积的, 同时也需要描述它的速度和加速状态。

由于 PSO 算法概念简单, 实现容易, 短短几年时间, PSO 算法便获得了很大的发展, 并在一些领域得到应用。目前已被“国际演化计算会议”(CEC)

列为讨论专题之一。

本文首先介绍 PSO 算法的标准版本, 然后根据一些测试结果对其参数设置进行讨论, 给出一些常用的测试函数, 随后讨论一些非标准的改进手段, 并给出一些与其他演化算法的比较, 以及 PSO 算法的应用例子。

### 2 PSO 标准版本

PSO 算法最初是为了图形化地模拟鸟群优美而不可预测的运动。人们通过对动物社会行为的观察, 发现在群体中对信息的社会共享有利于在演化中获得优势<sup>[4]</sup>, 并以此作为开发 PSO 算法的基础。通过加入近邻的速度匹配, 消除不必要的变量, 并考

收稿日期: 2001-11-02; 修回日期: 2001-12-06。

**作者简介:** 谢晓锋(1975—), 男, 湖南益阳人, 博士生, 从事带约束非线性规划问题的演化算法及半导体器件综合的研究; 杨之廉(1933—), 男, 上海人, 教授, 博士生导师, 从事 TCAD 系统的研究。

考虑多维搜索以及根据距离的加速,形成了 PSO 的最初版本<sup>[1]</sup>。之后, Shi 等<sup>[5]</sup> 引入惯性权重  $w$  来更好地控制开发(exploitation)和探索(exploration),形成了当前的标准版本。

### 2.1 算法原理

PSO 算法与其他演化算法相似,也是基于群体的,根据对环境的适应度将群体中的个体移动到好的区域,然而它不像其他演化算法那样对个体使用演化算子,而是将每个个体看作  $D$  维搜索空间中的一个没有体积的微粒(点),在搜索空间中以一定的速度飞行。这个速度根据它本身的飞行经验以及同伴的飞行经验进行动态调整。第  $i$  个微粒表示为  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , 它经历过的最好位置(有最好的适应值)记为  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ , 也称为  $p_{best}$ 。在群体所有微粒经历过的最好位置的索引号用符号  $g$  表示,即  $P_g$ , 也称为  $g_{best}$ 。微粒  $i$  的速度用  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  表示。对每一代,其第  $d$  维( $1 \leq d \leq D$ )根据如下方程变化<sup>[5]</sup>

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

$$x_{id} = x_{id} + v_{id} \quad (1b)$$

其中:  $w$  为惯性权重(inertia weight),  $c_1$  和  $c_2$  为加速常数(acceleration constants),  $\text{rand}()$  和  $\text{Rand}()$  为两个在  $[0, 1]$  范围内变化的随机函数。

此外,微粒的速度  $V_i$  被一个最大速度  $V_{max}$  所限制。如果当前对微粒的加速导致它在某维的速度  $v_{id}$  超过该维的最大速度  $v_{max,d}$ , 则该维的速度被限制为该维最大速度  $v_{max,d}$ 。

式(1a)的第1部分为微粒先前的速度;第2部分为“认知(cognition)”部分,表示微粒本身的思考;第3部分为“社会(social)”部分,表示微粒间的信息共享与相互合作<sup>[6]</sup>。

“认知”部分可由 Thorndike<sup>[7]</sup> 的“影响法则(law of effect)”解释,即一个得到加强的随机行为在将来更有可能出现。这里的行为即“认知”,并假设获得正确的知识是得到加强的,这样一个模型假定微粒被激励着去减小误差。

“社会”部分可由 Bandura<sup>[8]</sup> 的代理(vicarious)加强概念的解释。根据该理论的预期,当观察者观察到一个模型在加强某一行行为时,将增加它实行该行为的几率,即微粒本身的认知将被其他微粒所模仿。

PSO 算法的这些心理学假设是无争议的。在寻求一致的认知过程中,个体往往记住它们的信念,同时考虑同事们的信念。当个体察觉同事的信念较好

时,它将进行适应性地调整。

### 2.2 算法流程

标准 PSO 的算法流程如下:

Step1: 初始化一群微粒(群体规模为  $m$ ), 包括随机位置和速度;

Step2: 评价每个微粒的适应度;

Step3: 对每个微粒,将其适应值与其经历过的最好位置  $p_{best}$  作比较,如果较好,则将其作为当前的最好位置  $p_{best}$ ;

Step4: 对每个微粒,将其适应值与全局所经历的最好位置  $g_{best}$  作比较,如果较好,则重新设置  $g_{best}$  的索引号;

Step5: 根据方程(1)变化微粒的速度和位置;

Step6: 如未达到结束条件(通常为足够好的适应值或达到一个预设最大代数  $G_{max}$ ), 则返回 Step2。

## 3 算法参数分析

### 3.1 算法参数

PSO 参数包括: 群体规模  $m$ , 惯性权重  $w$ , 加速常数  $c_1$  和  $c_2$ , 最大速度  $V_{max}$ , 最大代数  $G_{max}$ 。

(1) 最大速度  $V_{max}$

$V_{max}$  决定当前位置与最好位置之间的区域的分辨率(或精度)。如果  $V_{max}$  太高,微粒可能会飞过好解;如果  $V_{max}$  太小,微粒不能在局部好区间之外进行足够的探索,导致陷入局部优值。

该限制有3个目的<sup>[6]</sup>:

- 1) 防止计算溢出;
- 2) 实现人工学习和态度转变;
- 3) 决定问题空间搜索的粒度。

(2) 权重因子

在 PSO 算法中有3个权重因子: 惯性权重  $w$ , 加速常数  $c_1$  和  $c_2$ 。

惯性权重  $w$  使微粒保持运动惯性,使其有扩展搜索空间的趋势,有能力探索新的区域。

加速常数  $c_1$  和  $c_2$  代表将每个微粒推向  $p_{best}$  和  $g_{best}$  位置的统计加速项的权重。低的值允许微粒在被拉回之前可以在目标区域外徘徊,而高的值则导致微粒突然的冲向或越过目标区域。

### 3.2 参数设置

如果没有后两部分,即  $c_1 = c_2 = 0$ , 微粒将一直以当前的速度飞行,直到到达边界。由于它只能搜索有限的区域,所以很难找到好解。

如果没有第1部分,即  $w = 0$ , 则速度只取决于微粒当前位置和其历史最好位置  $p_{best}$  和  $g_{best}$ , 速度本身没有记忆性。假设一个微粒位于全局最好位置

它将保持静止。而其他微粒则飞向它本身最好位置  $p_{best}$  和全局最好位置  $g_{best}$  的加权中心。在这种条件下, 微粒群将收缩到当前的全局最好位置, 更像一个局部算法。

在加上第 1 部分后, 微粒有扩展搜索空间的趋势, 即第 1 部分有全局搜索能力。这也使得  $w$  的作用为针对不同的搜索问题, 调整算法全局和局部搜索能力的平衡。

如果没有第 2 部分, 即  $c_1 = 0$ , 则微粒没有认知能力, 也就是“只有社会(social-only)”的模型<sup>[6]</sup>。在微粒的相互作用下, 有能力到达新的搜索空间。它的收敛速度比标准版本更快, 但对复杂问题, 则比标准版本更容易陷入局部最优值点。

如果没有第 3 部分, 即  $c_2 = 0$ , 则微粒之间没有社会信息共享, 也就是“只有认知(cognition-only)”的模型<sup>[6]</sup>。因为个体间没有交互, 一个规模为  $m$  的群体等价于运行了  $m$  个单个微粒的运行, 因而得到解的几率非常小。我们对一些函数的测试结果也验证了这一点。

早期的试验<sup>[1]</sup> 将  $w$  固定为 1.0,  $c_1$  和  $c_2$  固定为 2.0, 因此  $V_{max}$  成为唯一需要调节的参数, 通常设为每维变化范围的 10% ~ 20%<sup>[9]</sup>。

引入惯性权重  $w$  可消除对  $V_{max}$  的需要, 因为它们的作用都是维护全局和局部搜索能力的平衡。这样, 当  $V_{max}$  增加时, 可通过减小  $w$  来达到平衡搜索。而  $w$  的减小可使得所需的迭代次数变小<sup>[10]</sup>。从这个意义上看, 可以将  $v_{max, d}$  固定为每维变量的变化范围, 只对  $w$  进行调节。

对全局搜索, 通常的好方法是在前期有较高的探索能力以得到合适的种子, 而在后期有较高的开发能力以加快收敛速度。为此可将  $w$  设为随时间线性减小, 例如由 1.4 到 0<sup>[5]</sup>, 由 0.9 到 0.4<sup>[10, 11]</sup>, 由 0.95 到 0.2<sup>[12]</sup> 等。

Suganthan<sup>[12]</sup> 的实验表明,  $c_1$  和  $c_2$  为常数时可以得到较好的解, 但不一定必须为 2。Clerc<sup>[13]</sup> 引入收缩因子(constriction factor)  $K$  来保证收敛性

$$v_{id} = K[v_{id} + Urand() (p_{id} - x_{id}) + URand() (p_{gd} - x_{id})] \quad (2)$$

式中:  $K = \frac{2}{\hat{u}_2 - U - \frac{U^2}{U^2 - 4\hat{u}_1}}$ ,  $U = U + \hat{U}$ ,  $U >$

4。这对应于式(1a)中一种特殊的参数组合, 其中  $K$  即一种受  $U$  和  $\hat{U}$  限制的  $w$ , 而  $c_1 = KU$ ,  $c_2 = KU$ 。

这些参数也可以通过模糊系统进行调节。Shi 和 Eberhart<sup>[14]</sup> 提出一个模糊系统来调节  $w$ , 该系统

包括 9 条规则, 有两个输入和一个输出, 每个输入和输出定义了 3 个模糊集。一个输入为当前代的全局最好适应值, 另一个为当前的  $w$ ; 输出为  $w$  的变化。结果<sup>[14]</sup> 显示该方法能大为提高平均适应值。

此外, 群体的初始化也是影响算法性能的一个方面。Angeline<sup>[15]</sup> 对不对称的初始化进行了实验, 发现 PSO 只是略微受影响。

Ozcan 和 Mohan<sup>[16]</sup> 通过假设  $w = 1$ ,  $c_1$  和  $c_2$  为常数,  $p_{best}$  和  $g_{best}$  为固定点, 进行理论分析, 得到一个微粒随时间变化可以描述为波的运行, 并对不同的感兴趣的区域进行了轨迹分析。这个分析可以被 Kennedy 的模拟结果<sup>[17]</sup> 支持。一个寻求最优值位置的微粒尝试着操纵它的频率和幅度, 以捕获不同的波。 $w$  可以看作是修改了感兴趣的区域的边界, 而  $V_{max}$  则帮助微粒跳到另外一个波。

#### 4 常用测试函数<sup>[1, 11, 12, 15, 18, 19]</sup>

下面给出一些在文献中经常用到的测试函数。前两个是单峰函数, 另外几个为多峰函数, 其最优值均为或接近原点。以  $x$  代表一个实数类型的向量, 其维数为  $n$ , 而  $x_i$  为其中的第  $i$  个元素。

第 1 个为球面模型

$$f_0(x) = \sum_{i=1}^n x_i^2 \quad (3a)$$

这是个单峰函数, 用来区分局部优化器的优劣。

第 2 个为 Rosenbrock 函数, 也叫香蕉(Banana)函数, 是个经典的优化函数

$$f_1(x) = \sum_{i=1}^n (100(x_{i-1} - x_i^2)^2 + (x_i - 1)^2) \quad (3b)$$

第 3 个为 Rastrigrin 函数

$$f_2(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10) \quad (3c)$$

第 4 个为 Griewank 函数

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{i}\right) + 1 \quad (3d)$$

第 5 个为 Schaffer 函数<sup>[20]</sup>

$$f_4(x) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2} \quad (3e)$$

## 5 一些非标准方案

### 5.1 簇分解(cluster analysis) 方法<sup>[18]</sup>

对团体参考(reference groups) 和社会影响的研究<sup>[21]</sup> 表明, 人们基于所在团体的规范来选择他的观点和行为, 趋向于收敛在团体的平均层次上。

簇分解方法在微粒群体中选择一些微粒作为中心, 再将离它最近的  $N$  个微粒和中心微粒作为一簇, 然后计算每个簇的中心位置, 并以这些中心位置来替换  $p_{best}$  或者  $g_{best}$ 。

结果<sup>[18]</sup> 表明, 用簇中心替换  $p_{best}$  时, 测试函数 3(a) ~ 3(c) 的解得到了较好的改进, 而函数 3(d) ~ 3(e) 只是略微差一点。如果用全局的簇中心替换  $g_{best}$  时, 则除函数 3(c) 外, 结果都较差。

此外, 簇分解方法虽然使收敛速度有所加快, 但同时引入了一些附加计算, 通常需要正常 PSO 计算时间的 74% ~ 200%。

经验显示, 在个体被它们自己簇的中心所吸引时, PSO 相对有效; 如果被相邻簇的中心所吸引, 则一般不是很好。而总共需要多少簇, 这取决于问题空间的维数和局部优值点的数目。

### 5.2 利用选择(selection) 的方法<sup>[15]</sup>

PSO 算法只有一个隐含的弱选择, 即得到  $p_{best}$  的过程。而通常的演化算法, 选择方法用来重新定向搜索到那些相对较好的区域, 作为一个聚焦机制, 可以将有限的资源合理地分配。

加入选择可得到一种混合 PSO: 将每个个体的适应度, 基于其当前位置, 与  $k$  个其他个体进行比较, 并记下最差的一个点。群体再用这个记录排序, 最高的得分出现在群体的头部。在个体记分和排序过程中, 不考虑个人最好位置。一旦群体排完序, 群体中当前位置和速度最好的一半用来替换群体中差的另一半, 而不修改其个人最好位置  $p_{best}$ 。

对于每一代, 一半个体将移动到搜索空间中相对较优的位置, 这些移动的个体仍包含它们自己的个人最好点, 用来影响它们的下次位置。

对函数 3(a) ~ 3(d) 进行测试的结果<sup>[15]</sup> 表明, 混合方法提高了前 3 个函数的性能。这是由于选择方法加快了对当前较好区域的开发过程, 使得收敛速度较快, 但也增加了陷入局部解的可能性。

### 5.3 邻域算子(neighborhood operator)<sup>[12]</sup>

Angeline 的研究<sup>[22]</sup> 表明, 尽管 PSO 能比其他演化算法更快地得到相当有质量的解, 但当代数增加时, 并不能进行更精确的搜索。

为此, 可引入一个变化的邻域算子: 在优化的

初始阶段, 一个微粒的邻域就是它本身; 优化代数增加后, 邻域逐渐变大, 最后将包括所有的微粒。另外, 原来的全局历史最好值  $g_{best}$  将被微粒邻域的局部历史最好值  $l_{best}$  替代。

为得到邻域, 而计算候选个体与其他所有个体的距离, 其中对第 1 个个体的距离为  $dist[1]$ , 而最大距离为  $max-inst$ , 并定义一个与当前计算代数  $G_n$  有关的分数  $frac = 0.6 + 3.0G_n / G_{max}$ 。如果  $frac < 0.9$ , 且  $frac > dist[1] / max-inst$ , 则针对  $l_{best}$  进行搜索; 否则使用  $g_{best}$ 。

对函数 3(a) ~ 3(d) 进行测试的实验结果<sup>[12]</sup> 显示, 本方法的平均结果总是好于标准 PSO 的结果。

### 5.4 无希望 / 重新希望方法<sup>[13]</sup>

标准版本有 3 个权重因子, 这使得系统调整很自由, 但也为找到最好的参数组合带来难度。

Clerc 提出一个简化的 PSO 算法, 该算法只有一个公式和一个社会 / 信心参数。定义了一个无希望(No-hope) 收敛规则和一个重新希望(Re-hope) 方法, 以便不时地根据对目标函数的梯度估计和先前的初始化(意味着有初级的记忆) 重新初始化群体位置, 其中考虑了群体引力中心(Gravity center)。用该算法对函数 3(c) 和另一个二维函数进行测试, 得到了很好的结果<sup>[13]</sup>。

## 6 与其他演化算法的对比

### 6.1 与演化规划<sup>[23]</sup> 的对比

如果将式(1b) 看作一个变异算子, 则 PSO 算法与演化规划(EP) 很相似。然而, 在每一代, PSO 算法中的每个微粒只朝一些根据群体的经验认为是好的方向飞行, 而不象在演化规划中可通过一个随机函数变异到任何方向。也就是说, PSO 算法执行一种有“意识(Conscience)”的变异<sup>[10]</sup>。从理论上讲, 演化规划具有更多的机会在优化点附近开发, 而 PSO 则有更多的机会更快地飞到有更好解的区域, 如果“意识”能提供有用的信息。

### 6.2 与遗传算法<sup>[24]</sup> 的对比

遗传算法(GAs) 是模拟遗传选择和自然淘汰的生物进化过程的计算模型, 是由 Holland 于 1975 年提出的, 已广泛应用于计算机科学、优化调度、运输问题和组合优化等领域。

Kennedy 和 Spears<sup>[25]</sup> 利用多峰问题产生器<sup>[26]</sup> 对一个 PSO 的二进制版本<sup>[27]</sup> 和 3 种类型的 GAs(只有变异的 GA-m 和只有交叉的 GA-c 以及同时具有交叉和变异的 GA) 进行了研究。实验<sup>[25]</sup> 发现, PSO 基本都能更快地达到全局优值, PSO 基本不受问题

峰数增加的影响,受问题维数的影响也很小。

## 7 应用

PSO 算法已得到广泛应用,本文简要介绍一些例子。

PSO 最直接的应用或许就是多元函数的优化问题,包括带约束的优化问题<sup>[28]</sup>。如果所讨论的函数受到严重的噪音干扰而呈现非常不规则的形状,同时所求的不一定是精确的最优值,则 PSO 算法能得到很好的应用。

比如在半导体器件综合<sup>[29]</sup>方面,需要在给定的搜索空间内根据所希望的器件特性来得到符合要求的设计参数,而所能利用的器件模拟器通常得到的特性空间是高度非线性。我们用 PSO 替换 GAs 进行了计算,发现 PSO 能比 GAs 更快地找到较高质量的设计参数。

另外,还有一种应用更广泛的方法:简单而有效地演化的人工神经网络,不仅用于演化网络的权重,而且包括网络的结构<sup>[10, 30]</sup>。

作为一个演化神经网络的例子, PSO 算法已应用于分析人的颤抖<sup>[31]</sup>。对人颤抖的诊断,包括帕金森(Parkinson)病和原发性颤抖,是一个非常具有挑战性的领域。PSO 已成功地应用于演化一个用来快速和准确地辨别普通个体和有颤抖个体的神经网络,而网络的输入则为从一个活动变化记录系统中获得的归一化的移动振幅。

另一个应用例子是使用 PSO 对一个电气设备的功率反馈和电压进行控制<sup>[32]</sup>。这里,采用一种二进制与实数混合的 PSO 算法来决定对连续和离散的控制变量的控制策略,以得到稳定的电压。

此外, PSO 还在动态问题中得到应用<sup>[33]</sup>。

一般而言, PSO 与其他演化算法一样,能用于求解大多数优化问题。在这些领域中,最具潜力的有系统设计、多目标优化、分类、模式识别、信号处理、机器人技术应用、决策制定、模拟和证明等。例子包括模糊控制器设计、工作调度、实时机器人路径设计和图像分割等。

## 8 发展趋势与展望

对微粒群算法的研究,无论是在理论还是在实践方面都在不断发展中,已有的研究成果还相当分散。与相对鲜明的生物社会特性基础相比, PSO 的数学基础显得相对薄弱,缺乏深刻且具有普遍意义的理论分析。因此,对数学基础的研究非常重要,包括对不同搜索问题的收敛性、收敛速度估计、预防陷入局部优值和参数设置影响。

此外,还需要进行与其他优化技术的比较,更深入地了解其性能,并与其他技术(神经网络、模糊逻辑、复杂系统自组织<sup>[34]</sup>以及混沌理论等)相结合,以提高算法的计算性能。

算法的应用领域尚有待于进一步拓宽。

## 9 结论

PSO 算法是一种相当简单的算法,只需很少的代码和参数,但在各种问题的求解与应用中却展现了它的特点和魅力。该算法为人们提供了如下一种思路<sup>[1]</sup>:使智慧出现而不是努力强迫它;模拟自然而不是力图控制它;寻求使事情简单化而不是让它复杂。

目前我国针对 PSO 算法的研究还很少,而针对具体应用领域的研究则更为重要,以便促进我国高新技术的迅速发展。

## 参考文献(References):

- [1] Kennedy J, Eberhart R. Particle swarm optimization [A]. Proc IEEE Int Conf on Neural Networks[C]. Perth, 1995. 1942-1948.
- [2] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[A]. Proc 6th Int Symposium on Micro Machine and Human Science[C]. Nagoya, 1995. 39-43.
- [3] Millonas M M. Swarms Phase Transition and Collective Intelligence[M]. MA: Addison Wesley, 1994.
- [4] Wilson E O. Sociobiology: The New Synthesis[M]. MA: Belknap Press, 1975.
- [5] Shi Yuhui, Eberhart R. A modified particle swarm optimizer [A]. Proc IEEE Int Conf on Evolutionary Computation[C]. Anchorage, 1998. 69-73.
- [6] Kennedy J. The particle swarm: Social adaptation of knowledge[A]. Proc IEEE Int Conf on Evolutionary Computation[C]. Indianapolis, 1997. 303-308.
- [7] Thorndike E L. Animal Intelligence: Empirical Studies [M]. New York: MacMillan, 1911.
- [8] Bandura A. Social Foundations of Thought and Action: A Social Cognitive Theory[M]. New Jersey: Prentice-Hall, 1986.
- [9] Eberhart R, Shi Yuhui. Particle swarm optimization: Developments, applications and resources [A]. Proc IEEE Int Conf on Evolutionary Computation [C]. Seoul, 2001. 81-86.
- [10] Shi Yuhui, Eberhart R. Parameter selection in particle swarm optimization [A]. Proc of the 7th Annual Conf on Evolutionary Programming [C]. Washington DC, 1998. 591-600.
- [11] Shi Yuhui, Eberhart R. Empirical study of particle

swarm optimization[A]. Proc of the Congress on Evolutionary Computation [C]. Washington DC, 1999. 1945-1950.

- [12] Suganthan P N. Particle swarm optimiser with neighbourhood operator[A]. Proc of the Congress on Evolutionary Computation [C]. Washington DC, 1999. 1958-1962.
- [13] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization[A]. Proc of the Congress on Evolutionary Computation[C]. Washington DC, 1999. 1951-1957.
- [14] Shi Yuhui, Eberhart R. Fuzzy adaptive particle swarm optimization [A]. Proc IEEE Int Conf on Evolutionary Computation[C]. Seoul, 2001. 101-106.
- [15] Angeline P J. Using selection to improve particle swarm optimization [A]. Proc IEEE Int Conf on Evolutionary Computation[C]. Anchorage, 1998. 84-89.
- [16] Ozcan E, Mohan C. Particle swarm optimization: Surfing the waves[A]. Proc of the Congress on Evolutionary Computation [C]. Washington DC, 1999. 1939-1944.
- [17] Kennedy J. The behavior of particles [A]. Proc 7th Annual Conf on Evolutionary Programming [C]. Germany: Springer, 1998. 581-591.
- [18] Kennedy J. Stereotyping: Improving particle swarm performance with cluster analysis[A]. Proc IEEE Int Conf on Evolutionary Computation[C]. La Jolla, 2000. 1507-1512.
- [19] Eberhart R, Shi Yuhui. Comparing inertia weights and constriction factors in particle swarm optimization [A]. Proc IEEE Int Conf on Evolutionary Computation [C]. La Jolla, 2000. 84-88.
- [20] Davis L. Handbook of Genetic Algorithms[M]. New York: Van Nostrand Reinhold, 1991.
- [21] Sherif M, Sherif C W. Reference Groups: Exploration into Conformity and Deviation of Adolescents [M]. Chicago: Henry Regnery, 1964.
- [22] Angeline P J. Evolutionary optimization versus particle swarm optimization: Philosophy and performance difference[A]. Proc of the 7th Annual Conf on Evolutionary Programming [C]. Germany: Springer, 1998. 601-610.
- [23] Fogel L J, Walsh M J. Artificial Intelligence through Simulated Evolution [M]. Chichester: John Wiley,

1966.

- [24] Holland J H. Adaptation in Natural and Artificial Systems[M]. Ann Arbor: University of Michigan Press, 1975.
- [25] Kennedy J, Spears W M. Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator[A]. Proc IEEE Int Conf on Evolutionary Computation[C]. Anchorage, 1998. 78-83.
- [26] Jong K De, Potter M, Spears W. Using problem generators to explore the effects of epistasis[A]. Proc of the 7th Int Conf on Genetic Algorithms [C]. Morgan Kaufmann, 1997. 338-345.
- [27] Eberhart R, Kennedy J. Discrete binary version of the particle swarm algorithm[A]. Proc IEEE Int Conf on Systems, Man and Cybernetics[C]. Orlando, 1997. 4104-4108.
- [28] Ray T, Liew K M. A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimization problems [A]. Proc IEEE Int Conf on Evolutionary Computation[C]. Seoul, 2001. 75-80.
- [29] Xie Xiaofeng, Zhang Wenjun, Yang Zhilian. Incorporating knowledge in genetic algorithms for device synthesis[A]. 6th Int Conf on Solid-state and Integrated Circuit Technology [C]. Shanghai, 2001. 1143-1146.
- [30] Kenney J, Eberhart R, Shi Yuhui. Swarm intelligence[M]. San Francisco: Morgan Kaufmann, 2001.
- [31] Eberhart R, Hu Xiaohui. Human tremor analysis using particle swarm optimization[A]. Proc of the Congress on Evolutionary Computation [C]. Washington, 1999. 1927-1930.
- [32] Yoshida H, Kawata K, Fukuyama Y, et al. A particle swarm optimization for reactive power and voltage control considering voltage security assessment [J]. Trans of the Institute of Electrical Engineers of Japan, 1999, 119-B(12): 1462-1469.
- [33] Eberhart R, Shi Yuhui. Tracking and optimizing dynamic systems with particle swarms [A]. Proc IEEE Int Conf on Evolutionary Computation [C]. Hawaii, 2001. 94-100.
- [34] Prigogine I. Order through Fluctuation: Self-organization and Social System[M]. London: Addison-Wesley, 1976.