

文章编号: 100120920(2008)072077205

# 克服恋食行为的 PSO 算法改进研究

罗辞勇, 陈民铀

(重庆大学 a. 输变电设备及系统安全与新技术国家重点实验室, b. 电气工程学院, 重庆 400044)

**摘要:** 基本粒子群优化算法(PSO)存在易陷入局部极值的缺点. 为此, 研究鸟群迁徙觅食中的行为习惯, 以加强 PSO 的鸟群社会模型和对鸟群行为的模拟. 在所提出的改进算法中, 历史飞行速度在实际觅食中不作为判断因子, 只有发生位置重复时粒子才发生变异或摄动, 以此增强粒子群优化算法跳出局部最优解的能力. 实验结果表明, 新算法的全局搜索能力有了显著提高.

**关键词:** 粒子群算法; 飞行策略; 变异

**中图分类号:** TP18      **文献标识码:** A

## Improved PSO algorithm with overcoming behaviour of indulging in food

LUO Ci2yong, CHEN Min2you

(a. State Key Laboratory of Power Transmission Equipment and System Security and New Technology, b. School of Electrical Engineering, Chongqing University, Chongqing 400044, China. Correspondent: LUO Ci2yong, E2mail: luociyong@cqu.edu.cn)

**Abstract:** The basic PSO algorithm has the problem of local minimum. Therefore, the particle swarm optimisation (PSO) algorithm search capability is enhanced by the inclusion of new flight strategy and overcoming the behaviour of indulging in food based on the in2depth research of the social behaviour of flocks. In addition, the flight raidus is enlarged, and forepassed velocigy is not introduced to modify the particle. s position. When the particle. s position is repeated, a mutation operation will be introduced. The ability of breaking away from the local optimum is greatly improved by the mutation operation. The experimental results show that the new algorithm has advantages of convengence property.

**Key words:** PSO; Strategy of flight; Mutation

### 1 引言

粒子群优化算法(PSO)是由 Kenney 和 Eberhart<sup>[1]</sup>于 1995 年开发的一种典型的群体智能演化计算技术,其基本思想来源于对鸟群简化社会模型的研究及行为的模拟.在该算法中,寻优空间中的每个微粒没有质量和体积,其运动方向和运动速度受自身和群体的历史最优位置的影响,较好地协调了微粒本身与群体运动之间的关系<sup>[123]</sup>.

粒子群优化算法因其卓越的搜索速度,一经提出立即引起了众多学者的关注<sup>[412]</sup>.然而粒子群优化算法也存在易陷入局部最优的缺点.大量实验表明,粒子群算法无论是早熟还是达到全局最优解,

粒子群中的粒子都会聚集在搜索空间内的某一特定位置<sup>[12]</sup>.如果各个粒子的局部最优点与粒子群体全局最优点接近时,各个粒子的速度更新将主要由历史速度决定,粒子群呈现出强烈的趋同性<sup>0</sup>,这种趋同性加快了算法的搜索速度,但却减弱了群体开拓新的搜索空间的能力.

为了解决 PSO 的局部收敛问题,加快收敛速度,人们进行了大量的改进工作.如 Clerc 通过引入收敛因子  $k$  来保证收敛性.一些研究者还将进化算法中的进化机制引入微粒群算法,先后提出了带选择的 PSO,具有高斯变异的 PSO,以及具有繁殖和子种群的 PSO 等改进算法;考虑到微粒间合理高效的相互作用关系,又提出了簇分析 PSO,劳动分

收稿日期: 20070524; 修回日期: 200709217.

基金项目: 重庆市自然科学基金项目(CSTC2006BB2238).

作者简介: 罗辞勇(1973),男,安徽灵璧人,讲师,博士,从事智能控制理论及应用、阻抗成像等研究;陈民铀(1954),男,重庆人,教授,博士生导师,从事智能控制、数据建模等研究.

工 PSO 和协同 PSO 等; 还有人提出了进化 PSO, 多阶段 PSO 以及耗散 PSO 等<sup>[11]</sup>.

文献[12]提出了基于群体适应度方差自适应变异的粒子群优化算法. 算法在运行过程中, 根据群体适应度方差以及当前最优解的大小来确定当前最佳粒子的变异概率, 变异操作增强了粒子群优化算法跳出局部最优解的能力. 该算法的缺点是变异操作是在全局最佳粒子上进行的, 因而其跳出局部最优解的能力很有限.

PSO 算法的提出是对鸟群觅食过程中的迁徙和聚集的模拟, 也可以说是对社会心理学的模拟. 本文进一步考察鸟群迁徙觅食中的行为习惯, 以加强 PSO 的鸟群社会模型和对鸟群行为的模拟. 鸟群在迁徙中对目的地的确定, 是根据自己的最优经验和群体的最优经验作出判断的, 而历史飞行速度在实际觅食中不作为判断因子. 当食物充足时, 鸟群很容易产生恋食行为, 不会再飞行, 因而很可能失去寻求最理想觅食地的机会. 对此, 本文对 PSO 算法进行了改进: 1) 重新调整了飞行策略; 2) 鸟群发生恋食行为时, 引入变异操作. 仿真结果表明, 该算法计算剪度高, 收敛剪度快.

## 2 改进的 PSO 算法

### 2.1 基本 PSO 算法

在基本微粒群算法中, 由  $N$  个微粒组成的群体中的微粒在  $d$  维空间中的位置信息可表示为

$$x_i = (x_{i1}, x_{i2}, \dots, x_{id}), \quad i = 1, 2, \dots, N. \quad (1)$$

PSO 算法最早的表达式是由 Kenney 和 Eberhart<sup>[1]</sup>建立的, 随后, Shi 和 Eberhart<sup>[2]</sup>引入惯性权重对算法进行了改进. 在每次迭代过程中, 粒子的位置由下式确定:

$$v_i(t) = wv_i(t-1) + c_1r_1(p_i - x_i(t-1)) + c_2r_2(p_g - x_i(t-1)), \quad (2)$$

$$x_i(t) = v_i(t) + x_i(t-1). \quad (3)$$

其中:  $v_i$  为粒子飞行速度;  $w$  为惯性权重;  $c_1$  和  $c_2$  为加速度常数, 取正常数;  $r_1$  和  $r_2$  为在  $[0, 1]$  范围变化的随机数;  $p_i$  为粒子在历史中的最好位置;  $p_g$  为整个群体中的最好位置.

Shi 和 Eberhart 在式(2)的基础上, 根据迭代次数对参数  $w$  进行动态调整, 使得  $w$  随着代数的增加而逐渐减小, 称为自适应 PSO 算法 (APSO). 在 PSO 算法的搜索过程中, 随着代数的增加, 搜索区域会越来越小, 因此自适应 PSO 算法比基本 PSO 算法收敛速度更快, 效果更好. 具体算法可参见文献[3].

### 2.2 调整飞行策略

式(2)的第 1 项是  $wv_i(t-1)$ , 在 PSO 基本算法中, 该项视为历史速度对当前速度的影响, 通常选取

初始位置作为初始速度. Shi 和 Eberhart 等对  $w$  的性能和迭代变化规律进行了深入研究. 尽管从数学上对  $w$  的性能作了许多改进, 并取得了较好的效果, 但  $w$  的物理概念模糊, 其数值的选择需要依靠经验和复杂策略<sup>[428]</sup>. 本文重新考察鸟的飞行特点, 将去除这一项.

鸟在觅食过程中, 根据自己的最优经验和同伴的最优信息确定目标后, 飞行到达某个位置. 它在觅食时, 其运行速度为 0. 当它准备下次飞行时, 历史的飞行速度对它没有价值, 而对其飞行产生决定性的影响则是直观的眼前利益: 自己的最优经验和同伴的最优信息.

本文将式(2)和(3)改为

$$\begin{aligned} \$x = & c_1r_1(p_i - x_i(t-1)) + \\ & c_2r_2(p_g - x_i(t-1)), \end{aligned} \quad (4)$$

$$x_i(t) = x_i(t-1) + \$x, \quad (5)$$

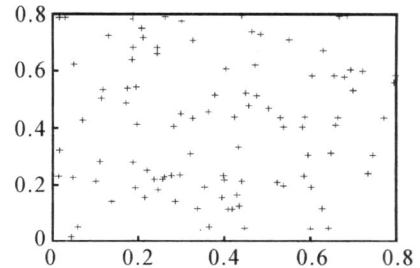
其中  $\$x$  为粒子飞行偏移量.

式(4)可以理解为粒子向自身和群体的历史最优位置飞行. 放弃了基本 PSO 算法中速度项, 从而也去掉了惯性权重  $w$  这个量. 式(5)的物理含义则很明确: 在上次位置开始, 叠加飞行距离, 到达新的位置.

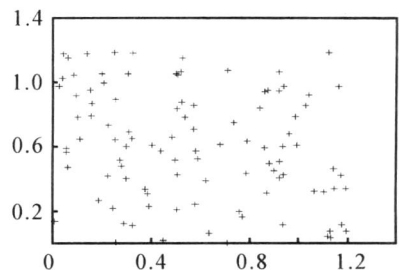
将式(4)中  $c_1$  和  $c_2$  重新定义为飞行系数, 它们将对飞行性能产生重要的影响. 下面以 100 个粒子从二维坐标  $(x_0, y_0) = (0, 0)$  向  $(x_1, y_1) = (1, 1)$  按照下式随机飞行:

$$\begin{cases} x_0 = x_0 + c_1(x_1 - x_0), \\ y_0 = y_0 + c_2(y_1 - y_0), \end{cases} \quad (6)$$

来分析飞行系数对飞行的影响. 随机飞行结果如图 1 所示.



(a)  $c=0.8$



(b)  $c=1.2$

图 1 不同飞行系数的随机飞行轨迹

在图1(a)中  $c=0.8$ , 没有一个粒子飞行到  $(x_1, y_1)$  附近. 在图1(b)中  $c=1.2$ , 有不少粒子飞行到  $(x_1, y_1)$  附近, 并有一些超越了  $(x_1, y_1)$ . 当  $c < 1$  时, 粒子的飞行能力有限, 飞行区域不覆盖目标点; 当  $c > 1$  时, 粒子飞行区域能够覆盖目标点, 可获取较大的飞行空间. 当然  $c > 1$  的值也不能太大, 否则不会向目标点集中, 从而失去意义. 因此  $c$  在  $[1, 2]$  区间内比较合适.

### 2.3 克服恋食行为

基本 PSO 算法无论是早熟还是达到全局最优解, 粒子群中的粒子都会聚集在搜索空间内的某一特定位置. 粒子总是向全局最优或个体最优飞行, 经过一定次数的迭代后, 可能发生全局最优、个体最优与所有粒子都相同的情况. 这时按式(2)和(3)迭代, 粒子的位置将不再发生变化. 按式(4)和(5)迭代也不能避免这个问题.

从鸟群的觅食角度看, 如果停止飞行时, 依然有足够多的食物够鸟群分享, 它们自然会留恋在原地, 而不能迈开脚步, 这种行为称为恋食. 发生恋食行为时, 鸟群所搜索的场地可能并不是最优的, 但鸟群不愿意离开. 鸟群如何才能离开呢? 从社会学角度来看, 当食物消耗完毕后, 鸟儿自然会离开原地, 重新飞翔. 一部分鸟儿会在原地周围游移, 寻觅可能遗留的食物; 另一部分则开始漫无目标地探索, 像哥伦布一样去寻找新大陆.

如果规定在一个位置只有一份食物, 则任何一只鸟儿发生觅食后, 其他的鸟儿将不能再觅食. 即便鸟儿总是向最优的地方飞行, 若目的地没有食物, 鸟儿就会改变飞行目标. 这样就会克服恋食行为.

在优化问题中, 一个位置计算一次称为 1 次觅食行为, 发生计算后这个位置就没有食物了, 下次也就没有必要再计算了. 粒子按照式(4)和(5)飞行, 首先判断位置是否重复, 如果从没有粒子到达, 则进行觅食; 如果重复, 则改变飞行目的后再次觅食. 如果要严格实现这种想法, 需要记录所有到达过的位置, 并且遍历所有的历史位置来判断是否重复, 这个工作量是非常巨大的. 因此在算法中, 可简化为只与全局历史最优和个体历史最优进行比较, 判断位置是否重复. 判断式为

$$+p_g - x_i < D \quad i = 1, 2, \dots, N;$$

$$\text{或} \quad +p_j - x_i < D \quad i, j = 1, 2, \dots, N. \quad (7)$$

其中  $D$  是重复半径. 最大计算次数为  $N @ (N + 1)$  次.

如果式(7)成立, 则判定粒子位置重复成立, 并按如下步骤处理:

1) 令  $p = \text{rand}$  为  $[0, 1]$  范围的随机数.

2) 如果  $p \leq p_m$ , 则对粒子位置进行变异, 模拟鸟儿随机飞行.  $p_m$  是变异比例, 小于 1.

3) 如果  $0 < p < p_m$ , 则让粒子在当前位置按下式进行摄动:

$$x_i(t) = x_i(t) + K(0.5 - \text{rand}). \quad (8)$$

其中:  $K$  是摄动因子,  $\text{rand}$  是在  $[0, 1]$  范围变化的随机数.

第 3) 步是模拟鸟儿在当前位置徘徊的行为.

当发生粒子群中的粒子聚集在搜索空间内的某一特定位置时, 上述操作将会迫使粒子离开特定位置, 从而避免陷入局部最优. 克服恋食行为的策略也会节约计算成本和提高计算效率.

### 2.4 完整的改进算法

综合上述改进, 完整的改进 PSO 算法 (IPSO) 为:

Step1: 初始化. 设定种群规模  $N$  和迭代次数  $N_t$ . 用预定义的值域随机初始化粒子位置  $x_i$ , 个体最优位置  $p_i = x_i$ , 并计算适应度. 迭代计数器  $t = 0$ .

Step2: 评价.  $t = t + 1$ . 评价当前种群中所有粒子的适应度, 从中选择最优适应度粒子. 该粒子对应的位置就是全局最优位置  $p_g$ , 所对应的适应度就是全局最优适应度. 若全局最优适应度满足收敛条件, 则终止迭代.

Step3: 如果  $t \geq N_t$ , 则终止迭代.

Step4: 飞行和克服恋食. 按式(4)和(5)计算新粒子. 用式(7)判断是否位置重复. 如果重复, 则按 2.2 节的方法来处理 (部分变异, 部分摄动). 这个过程完成后将产生新一代粒子.

Step5: 计算新一代粒子的适应度. 将计算出来的值与个体最优的适应度进行比较, 如果好, 则将其作为个体最好适应度, 其位置即为个体最优位置. 转 Step2.

## 3 收敛性分析

PSO 寻优算法求解空间问题需满足如下假设:

假设 1<sup>[13]</sup> 问题  $P$  的可行域  $\delta$  为  $R^n$  中的有界闭区域, 目标函数  $f(x)$  是区域  $\delta$  上的连续函数.

定义 1<sup>[13]</sup> 设  $\{X(k)\}$  是由算法  $M$  产生的种群序列, 其中  $x^*(k) \in X(k)$  为第  $k$  代种群中的最优个体. 若存在  $f(x^*(k)) \leq f(x^*(k-1))$ , 且序列中的某一点  $x^*(N)$  本身或者序列有一个极限  $x^*$  是问题  $P$  的一个极小点, 则称算法  $M$  是局部收敛的.

引理 1 若问题  $P$  存在局部极小点  $x^*$ , 则

$$\lim_{t \rightarrow \infty} p_g(t) = x^*.$$

证明 因为  $p_g(t)$  为迭代过程中  $t$  时刻的微粒群最优个体, 则有  $f(p_g(t)) \leq f(p_g(t-1))$  成立.

$$x_i(t+1) = x_i(t) + c_1 r_1(p_i(t) - x_i(t)) +$$

$$c_2 r_2 (p_g(t) - x_i(t)). \quad (9)$$

对于种群最优个体有  $p_g(t) = p_i(t)$ , 于是式(9) 变为

$$x_i(t+1) = x_i(t) + (c_1 r_1 + c_2 r_2)(p_g(t) - x_i(t)). \quad (10)$$

设  $B = c_1 r_1 + c_2 r_2$ , 整理式(10) 得

$$x_i(t+1) - x_i(t) = B(p_g(t) - x_i(t)). \quad (11)$$

上式表明,  $x(t)$  将逐渐逼近种群最优点  $p_g(t)$ <sup>[13, 14]</sup>.

假设  $\lim_{t \rightarrow \infty} p_g(t)$  不是局部最小点, 则存在  $r > 0$ , 使得当  $+x - p_g(t) + [r]$  时, 有  $f(x) < f(p_g(t))$ . 因为  $f(x)$  连续, 由式(11) 在  $x(t)$  逼近  $p_g(t)$  的过程中, 必然存在  $t_c < t$ , 使得  $+x(t_c) - p_g(t) + [r]$ , 即有  $f(x(t_c)) < f(p_g(t))$  成立. 这与已知相矛盾, 所以假设不成立. 故  $\lim_{t \rightarrow \infty} p_g(t)$  为局部极小点.  $t$

定理 1 IP SO 算法是局部收敛的.

证明 对于种群, 最优微粒存在序列  $f(p_g(t))$   $[f(p_g(t-1))]$ ,  $[f(p_g(0))]$ , 并且由引理 1 可知  $\lim_{t \rightarrow \infty} p_g(t) = x^*$  ( $x^*$  是局部极小点), 又由定义 1 可知, IP SO 是局部收敛的.  $t$

引理 2 在 IP SO 算法中,  $r_1, r_2 \sim N(0, 1)$ , 则有  $\$x \sim N(L_d, R_d)$ .

证明 设

$$\begin{aligned} \leq_1 &= c_1(p_i - x_i(t-1)), \\ \leq_2 &= c_2(p_g - x_i(t-1)), \end{aligned}$$

则式(4) 可整理为

$$\$x = \leq_1 r_1 + \leq_2 r_2.$$

对于 IP SO 算法, 在粒子飞行过程中, 当发生了恋食行为后, 进行部分变异和部分摄动的处理, 可使  $\leq_1$  和  $\leq_2$  为非 0 变量. 由于  $r_1, r_2 \sim N(0, 1)$ , 由文献 [13] 可知,  $\$x \sim N(L_d, R_d)$ .  $t$

定理 2 设  $\{X(k)\}$  是由 IP SO 算法产生的种群序列, 其中  $x^*(k) \in X(k)$  为第  $k$  代种群中的最优个体, 即  $x^*(k) = \arg \min_{i \in \{1, \dots, L\}} f(x_i(k))$ , 如果问题 P 中的目标函数和可行域满足假设 1, 则有

$$P\{\lim_{k \rightarrow \infty} f(x^*(k)) = f^*\} = 1,$$

即种群以概率为 1 收敛于全局最优解.

证明 由引理 2 可知,  $\$x \sim N(L_d, R_d)$ . 前面分析表明, IP SO 是一个带有变异和摄动的进化策略算法, 由文献 [13, 15, 16] 可知, 定理 2 成立 (详细证明参见文献 [13, 15, 16]).  $t$

#### 4 优化实例

为分析 IP SO 算法的收敛速度和全局搜索性能, 本文选取 3 个 Benchmark 优化问题进行分析, 表 1 给出了这些 Benchmark 函数的定义. 测试时, 各函

数的搜索范围均为  $-10 < x_i < 10$ .

表 1 Benchmark 测试函数<sup>[13, 14]</sup>

函数名称	数学公式
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigrin	$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$

选择基本 PSO 和 AP SO, 与 IP SO 进行性能比较. 所有实验的维数  $d = 2$ , 种群规模  $N = 50$ , 最大迭代次数  $N_t = 100$ , 收敛条件  $f(x) < 1.0e210$ . 若超过  $N_t$  次还没收敛, 则视为搜索失败. 基本 PSO 中  $w$  取 0.8, 在 AP SO 中选择  $w$  在  $[0.95, 0.4]$  之间随代数线性递减, 飞行系数  $c_1 = c_2 = 0.96$ . 在 IP SO 中, 飞行系数  $c_1 = c_2 = 1.43$ , 重复半径  $D = 1.0e27$ , 变异比例  $p_m = 0.5$ , 摄动因子  $K = 0.002$ . IP SO, 基本 PSO 及 AP SO 各进行 100 次并进行对比分析, 结果见表 2.

表 2 Benchmark 测试函数对比测试结果

函数	算法	平均迭代次数	最快迭代次数	搜索成功率 / %	最优解	最劣解
$f_1$	PSO	91	65	88	$1.5e2012$	$1.1e209$
	AP SO	54	45	100	$3.4e2013$	$9.8e2011$
	IP SO	13	11	100	$8.0e2013$	$9.6e2011$
$f_2$	PSO	100	93	1	$6.6e2011$	$1.2e206$
	AP SO	92	72	95	$2.4e2013$	$9.7e2011$
	IP SO	31	15	100	$9.3e2013$	$9.8e2011$
$f_3$	PSO	100	97	4	$4.2e2012$	$4.2e2007$
	AP SO	83	73	100	$9.4e2013$	$9.9e2011$
	IP SO	30	16	100	$3.1e2013$	$9.9e2011$

由表 2 可见, 使用线性下降惯性权重的 AP SO 比基本 PSO 具有更快的收敛速度和搜索成功率, AP SO 在 Rosenbrock 函数上的搜索成功率是 95%, 基本 PSO 只有 1%, 而 IP SO 在 3 个函数上均获得了 100% 搜索成功率. 从收敛速度上看, IP SO 算法对于所有函数均能取得较快的收敛速度, 最快迭代次数均在 20 代以下, 对于 Sphere 函数平均迭代次数更是达到了 13. 测试结果表明, IP SO 算法在发生恋食行为时给出了相应的调整策略, 与 AP SO 和基本 PSO 相比, 显著提高了算法的收敛速度.

IP SO 算法克服了趋同性, 能避免陷入局部最优点. 但不能保证最终所有粒子都能到达最优点. 图 2 为 IP SO 在 Rastrigrin 函数上迭代结束后的粒子位置, 可以看出, 有较多的粒子没落在  $(0, 0)$  理想位置

上,而是散落在周围.所以,IPSO 算法体现的是群体的最优,而非个体的最优.

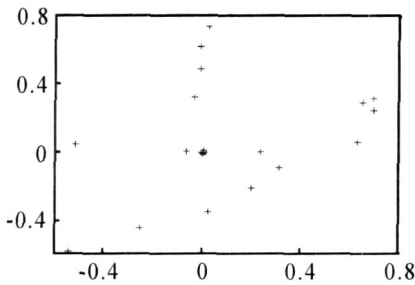


图2 迭代结束后的粒子位置

## 5 结 论

粒子群优化算法存在容易陷入局部最优的缺点.如果各个粒子的局部最优点同粒子群体全局最优点接近时,粒子群呈现出强烈的趋同性,这种趋同性加快了算法的搜索速度,但同时也减弱了群体开拓新的搜索空间的能力.根据鸟群的飞行特点和克服恋食的办法,本文提出了改进的PSO算法.仿真结果证明了该算法的有效性.

## 参考文献(References)

[1] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc of IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.

[2] Shi Y, Eberhart R. A modified particle swarm optimizer [C]. IEEE World Congress on Computational Intelligence. Anchorage, 1998: 692-73.

[3] Angeline P J. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences [C]. Proc of the 7th Annual Conf on Evolutionary Programming. Germany, 1998: 602-610.

[4] Kennedy J. The particle swarm: Social adaptation of knowledge [C]. Proc of 1997 Int Conf on Evolutionary Computation. Indianapolis, 1997: 3032-308.

[5] Kennedy J. Stereotyping: Improving particle swarm performance with cluster analysis [C]. Evolutionary Computation. La Jolla, 2000: 1507-1512.

[6] Eberhart R, Shi Y. Comparison between genetic algorithms and particle swarm optimization [J]. Lecture Notes in Computer Science, 1998, 22(3): 611-616.

[7] Shi Y, Eberhart R. Empirical study of particle swarm optimization [C]. Evolutionary Computation, Washington, 1999: 194-195.

[8] Zhang L H, Hu S. A new approach to improve particle swarm optimization [J]. Lecture Notes in Computer Science, 2003, 27(23): 134-139.

[9] 王建辉, 徐林, 闫勇亮, 等. 改进粒子群算法及其对热连轧机负荷分配优化的研究 [J]. 控制与决策, 2005, 20(12): 1379-1383.  
(Wang J H, Xu L, Yan Y L, et al. Improved PSO and its application to load distribution optimization of hot strip mills [J]. Control and Decision, 2005, 20(12): 1379-1383.)

[10] 赵波, 曹一家. 电力系统无功优化的多智能体粒子群优化算法 [J]. 中国电机工程学报, 2005, 25(5): 127.  
(Zhao B, Cao Y J. A multi-agent particle swarm optimization algorithm for reactive power optimization [J]. Proc of the CSEE, 2005, 25(5): 127.)

[11] 汪镭, 康琦, 吴启迪. 基于形式化描述的逻辑分层延迟 PSO 算法及应用 [J]. 控制与决策, 2005, 20(6): 612-615.  
(Wang L, Kang Q, Wu Q D. Logical layered and delayed PSO algorithm and application based on formalization description [J]. Control and Decision, 2005, 20(6): 612-615.)

[12] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法 [J]. 电子学报, 2004, 32(3): 416-420.  
(Lv Z S, Hou Z R. Particle swarm optimization with adaptive mutation [J]. Acta Electronica Sinica, 2004, 32(3): 416-420.)

[13] 赫然, 王永吉, 王青, 等. 一种改进的自适应逃逸微粒群算法及实验分析 [J]. 软件学报, 2005, 16(12): 2036-2044.  
(He R, Wang Y J, Wang Q, et al. An improved particle swarm optimization based on self-adaptive escape velocity [J]. J of Software, 2005, 16(12): 2036-2044.)

[14] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.

[15] 李宏, 唐焕文, 郭崇慧. 一类进化策略的收敛性分析 [J]. 运筹学学报, 1999, 3(4): 79-83.  
(Li H, Tang H W, Guo C H. The convergence analysis of a class of evolution strategies [J]. Operations Research Transactions, 1999, 3(4): 79-83.)

[16] 郭崇慧, 唐焕文. 演化策略的全局收敛性 [J]. 计算数学, 2001, 23(1): 105-110.  
(Guo C H, Tang H W. Global convergence properties of evolution strategies [J]. Mathematica Numerica Sinica, 2001, 23(1): 105-110.)