第 25 卷 第 9 期
Vol. 25 No. 8

控 制 与 决 策
Control and Decision

2010 年 9 月
Sep. 2010

文章编号: 1001-0920(2010)09-1333-05

基于进化停滞周期的局部变异PSO算法及其收敛性分析

曾 华[a], 吴耀华[a,b]

(山东大学 a. 控制科学与工程学院，b. 现代物流研究中心，济南 250061)

摘 要: 为了克服粒子群优化算法容易陷入局部最优而发生早熟收敛的问题, 提出一种基于进化停滞周期的局部变异粒子群优化算法. 算法引入进化停滞周期和近期全局最优位置的概念, 使粒子的飞行受近期全局最优位置影响, 并在种群进化停滞时对随机选中的局部粒子执行变异操作, 增加种群多样性, 扩大搜索范围, 提高求解质量. 算法用种群进化停滞周期代替多样性度量, 避免了多样性计算引起的高计算复杂度. 对于几个常用基准函数的仿真结果验证了算法的合理性和有效性.

关键词: 多峰优化；粒子群优化算法；进化停滞周期；局部变异

中图分类号: TP18 **文献标识码:** A

PSO algorithm with local mutation in evolution stagnation cycle and its convergence analysis

ZENG Hua[a], *WU Yao-hua*[a,b]

(a. School of Control Science and Engineering，b. The Logistics Institute，Shandong University，Ji'nan 250061，China. Correspondent：ZENG Hua，E-mail：zenghua1981@163.com)

Abstract: To overcome the premature of particle swarm optimization(PSO) algorithm, an algorithm called particle swarm optimization with local mutation in evolution stagnation cycle(LSPSO) is presented. Concepts of evolution stagnation cycle and recent global best position are proposed, so that particles are influenced by recent global best position instead of global best position, and a random local mutation operation of particles is taken when the evolution of population stagnates. These strategies enrich the diversity of population, extend the search space, and improve the quality of solution. Instead of computing the diversity of population, evolution stagnation cycle is used for lower computing complexity. The simulation results show the reasonability and effectiveness of the algorithm.

Key words: Multimodal optimization；Particle swarm optimization；Evolution stagnation cycle；Local mutation

1 引 言

粒子群优化(PSO)算法是Kennedy等[1,2]在鸟群捕食行为研究的基础上提出的一种群体智能进化算法. 由于其原理简单且并行搜索能力强, 近年来成为自然计算领域新的研究热点, 并广泛应用于复杂函数优化、神经网络训练、参数优化和资源调度等领域[3-6].

与遗传算法等智能优化算法类似, PSO算法也存在早熟收敛的问题. 为平衡其进化过程中的全局搜索和局部搜索, 有关学者在多方面做了改进, 如: 与其他优化算法相结合[6]、动态调整惯性权重[7]、通过控制种群多样性提高算法性能[8]、划分粒子分群并执行多种搜索策略[9]等. 虽然这些工作在一定程度上提高了PSO算法的全局搜索能力, 但仍存在搜索后期种群多样性丧失, 或在保持种群多样性的同时增加大量计算的不足. 针对这一问题, 本文提出一种基于进化停滞周期的局部变异粒子群优化(LSPSO)算法. 算法在种群连续若干代进化停滞时, 将随机选中的部分粒子驱散到新的搜索区域, 继续全局范围搜索. 通过几个Benchmark函数对算法性能进行验证. 仿真结果表明该方法能够有效增强种群多样性, 避免早熟收敛, 提高算法的全局搜索能力.

2 基于进化停滞周期的局部变异PSO算法

2.1 基于进化停滞周期的局部变异策略

针对文献[7]中标准PSO算法早熟收敛的缺陷,

收稿日期: 2009-07-12；**修回日期:** 2009-09-15.

基金项目: 国家自然科学基金项目(50175064)；山东大学研究生自主创新基金项目(31400070613065).

作者简介: 曾华(1981−), 女, 黑龙江肇东人, 博士生, 从事数据挖掘、组合优化算法的研究；吴耀华(1963−), 男, 沈阳人, 教授, 博士生导师, 从事物流系统建模与仿真、控制决策等研究.

在本文提出的LSPSO算法中引入近期最优位置和进化停滞周期的概念, 当进化停滞达到一定程度时, 将随机选取的一部分粒子驱散到新的搜索空间, 以提高全局寻优能力.

首先, 定义近期全局最优位置, 用于记录种群最近一个"时期"搜索到的全局最优位置, 记为 p'_g. LSPSO算法通过 p'_g 吸引粒子, 并继续用 p_g 记录整个搜索过程中种群的最优位置. 当粒子因集中到某一小范围区域而停止进化时, 认为这一"时期"搜索任务完成, 并开始下一"时期"的搜索工作. 随着进化代数的增加和搜索"时期"的推进, p'_g 将记录不同"时期"的全局最优位置. LSPSO算法的速度和位置更新公式为

$$v_i(t+1)=w(t)v_i(t)+c_1r_1(p_i(t)-x_i(t))+ c_2r_2(p'_g(t)-x_i(t)), \tag{1}$$

$$x_i(t+1)=x_i(t)+v_i(t+1), \tag{2}$$

其中惯性权重 w 随迭代次数线性递减[7], 即

$$w(t)=w_{\max}-(w_{\max}-w_{\min})t/T, \tag{3}$$

$w_{\min}$ 和 $w_{\max}$ 分别为惯性权重变化的最小值和最大值. 近期全局最优位置 p'_g 的更新公式为

$$p'_g(t+1)=\begin{cases} x^*(t+1),\ x^*(t+1)\text{ 优于 }p'_g(t);\\ p'_g(t),\ \text{otherwise}. \end{cases} \tag{4}$$

其中 $x^*(t+1)$ 为第 $t+1$ 次迭代中种群最优粒子位置. 历史最优位置 p_g 的更新公式为

$$p_g(t+1)=\begin{cases} p'_g(t+1),\ p'_g(t+1)\text{ 优于 }p_g(t);\\ p_g(t),\ \text{otherwise}. \end{cases} \tag{5}$$

其次, 在LSPSO算法中定义进化停滞周期, 并以此作为判断算法早熟收敛的依据, 以便在适当的时候对部分粒子执行变异操作, 保持种群多样性.

定义 1 设函数 f 是最小化优化问题的适应度函数, $p_g(t)$ 是种群在第 t 次迭代得到的历史最优位置, 对于给定的非负数 $\delta\geqslant 0$, 若有

$$f(p_g(t))-f(p_g(t+1))\leqslant\delta \tag{6}$$

成立, 则称种群在第 $t+1$ 次进化中对于 δ 半径进化停滞, 称 δ 为进化停滞半径.

类似地, 可以定义最大化优化问题中的进化停滞, 在此不再赘述.

定义 2 对于给定的进化停滞半径 $\delta\geqslant 0$, 若种群在第 t 次迭代前的连续 n 次迭代中均对 δ 进化停滞, 则称 n 为种群在第 t 次迭代的进化停滞周期.

由定义2得到种群进化停滞周期的更新公式为

$$n(t+1)=\begin{cases} n(t)+1,\ f(p_g(t))-f(p_g(t+1))\leqslant\delta;\\ 0,\ f(p_g(t))-f(p_g(t+1))>\delta. \end{cases} \tag{7}$$

在鸟群觅食的自然行为中, 当所有个体因发现一块食物而集中到一处时, 必须将部分甚至全部个体驱散到其他领域, 才有机会获取更好的食物. 为了在保留历史经验的同时进一步拓展新的探索领域, 当种群进化停滞周期达到一定值时, 对部分粒子执行变异操作, 以提高种群的多样性.

对于给定进化停滞周期阈值 N, 若在第 t 代有 $n(t)\geqslant N$ 成立, 则对每一粒子执行以下操作:

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

2) 若 $0<p\leqslant p_m$, 则对粒子位置进行变异, 其中 p_m 是变异因子, $0<p_m<1$.

2.2 LSPSO算法的收敛性分析

假设 1 问题 P 的可行域 Ω 为 R^n 中的有界闭区间, 目标函数 $f(x)$ 是区域 Ω 上的连续函数[10,11].

定义 3 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 $f(x^*(k))\leqslant f(x^*(k-1))$, 且序列中某一点 $x^*(N)$ 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[10,11].

引理 1 若问题 P 存在局部极小点 x^*, 则

$$\lim_{t\to\infty}p_g(t)=x^*. \tag{8}$$

证明 因为 $p_g(t)$ 为迭代过程中 t 时刻种群中最优粒子, 有 $f(p_g(t))\leqslant f(p_g(t-1))$ 成立. 根据算法速度和位置更新公式, 有

$$x_i(t+1)= x_i(t)+w(t)v_i(t)+c_1r_1(p_i(t)- x_i(t))+c_2r_2(p_g(t)-x_i(t)). \tag{9}$$

由于 $v_i(t)=x_i(t)-x_i(t-1)$, 有

$$x_i(t+1)= x_i(t)+w(t)(x_i(t)-x_i(t-1))+c_1r_1(p_i(t)- x_i(t))+c_2r_2(p_g(t)-x_i(t)). \tag{10}$$

对于种群最优个体有 $p_g(t)=p_i(t)$. 于是, 式(10)变为

$$x_i(t+1)= x_i(t)+w(t)(x_i(t)-x_i(t-1))+ (c_1r_1+c_2r_2)(p_g(t)-x_i(t)). \tag{11}$$

设 $\beta=c_1r_1+c_2r_2$, 式(11)整理得

$$x_i(t+1)= x_i(t)+w(t)(x_i(t)-x_i(t-1))+ \beta(p_g(t)-x_i(t)). \tag{12}$$

式(12)表明, $x(t)$ 将逐渐逼近种群最优点 $p_g(t)$.

假设 $\lim_{t\to\infty}p_g(t)$ 不是局部极小点, 则存在 $r>0$, 使得当 $\|x-p_g(t)\|\leqslant r$ 时, 有 $f(x)<f(p_g(t))$. 因为 $f(x)$ 连续, 在 $x(t)$ 逼近 $p_g(t)$ 的过程中, 必存在 $t'<t$, 使得 $\|x(t')-p_g(t)\|\leqslant r$, 即有 $f(x(t'))<f(p_g(t))$ 成立. 这与已知相矛盾, 所以假设不成立. 故 $\lim_{t\to\infty}p_g(t)$ 为局部极小值. □

定理 1 LSPSO算法是局部收敛的.

证明　对于种群最优粒子 i, 有 $p_g(t) = p_i(t)$, 存在序列 $f(p_g(t)) \leqslant f(p_g(t-1)) \leqslant \cdots \leqslant f(p_g(0))$. 由引理 1 可知, $\lim\limits_{t\to\infty} p_g(t) = x^*$($x^*$ 是局部极小点), 又由定义 3 可知, LSPSO 算法是局部收敛的. □

引理 2　设 $\Delta x_i = v_i, r_1, r_2 \sim N(0,1)$, 则有 $\Delta x \sim N(\mu_d, \sigma_d)$, 其中 d 是空间维数.

证明　由式 (1) 可知

$$\Delta x_i = wv_i + c_1r_1(p_i - x_i) + c_2r_2(p_g - x_i). \tag{13}$$

设

$$\phi_1 = wv_i,\ \phi_2 = c_1(p_i - x_i(t-1)),$$
$$\phi_3 = c_2(p_g - x_i(t-1)),$$

有

$$\Delta x = \phi_1 + \phi_2 r_1 + \phi_3 r_2. \tag{14}$$

对于 LSPSO 算法, 在粒子飞行过程中, 当进化停滞时, 局部变异操作和近期全局最优位置的更新操作及时给粒子一个较大的速度冲量, 使 ϕ_1, ϕ_2 和 ϕ_3 为非 0 变量, 由 $r_1, r_2 \sim N(0,1)$ 可知 $\Delta x \sim N(\mu_d, \sigma_d)$. □

定理 2　设 $\{X(k)\}$ 是由 LSPSO 算法生成的种群序列, $x^*(k) \in X(k)$ 为第 k 代种群中的最优个体, 即 $x^*(k) = \arg\min\limits_{1\leqslant i\leqslant\mu} f(x_i(k))$. 如果问题 P 中的目标函数和可行域满足假设 1, 则有

$$P\{\lim_{k\to\infty} f(x^*(k)) = f^*\} = 1, \tag{15}$$

即种群以概率 1 收敛于全局最优位置.

证明　由引理 2 可知 $\Delta x \sim N(\mu_d, \sigma_d)$. 前面分析表明, LSPSO 是一个带有局部变异操作的进化策略算法, 由文献 [10,11] 可知, 定理 2 成立. □

3　实验及结果分析

3.1　实验设计

为了分析 LSPSO 算法的收敛稳定性和全局搜索性能, 本文应用 LSPSO 算法对选取的 7 个 Benchmark 函数进行仿真, 并与惯性权重线性下降的标准粒子群算法[7] (LWPSO), 动态惯性权重及速度钳位粒子群算法[9] (DPSO), 自适应粒子群算法[12] (FAPSO) 以及克服恋食行为的粒子群算法[13] (IPSO) 作对比. 测试函数的函数形式、搜索范围、理论极值和优化目标精度如表 1 所示.

在 Matlab 7.0 环境中编写 m 文件形式的测试程序. 为增强可比性, 实验中设置各种算法的种群规模 $s_{\text{size}} = 20$, 迭代次数 $T = 500$, 分别独立运行上述算法 30 次 ($K = 30$). 参加测试的各算法参数取值如表 2 所示. 实验分别从收敛精度、收敛速度和成功率等方面对算法性能进行评估.

3.2　实验结果及分析

表 3 为 LWPSO, DPSO, FAPSO, IPSO 和 LSPSO 的具体实验结果.

3.2.1　算法收敛精度分析

从适应度的均值、最优值、最差值和标准差等指

表 1　用于测试改进算法的 Benchmark 函数

Name and code	Formula	Range$[x_{\min}, x_{\max}]$	Optimal f	Criterion
Sphere f_1	$f(X) = \sum_{i=1}^{p} x_i^2$	$x_i \in [-100,\ 100]^3$	0	1.00E-20
Rosenbrock f_2	$f(X) = \sum_{i=1}^{p-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$x_i \in [-100,\ 100]^3$	0	1.00E+02
Rastrigin f_3	$f(X) = 10p + \sum_{i=1}^{p} (x_i^2 - 10\cos(2\pi x_i))$	$x_i \in [-100,\ 100]^3$	0	2.00E+00
Griewangk f_4	$f(X) = 1 + \sum_{i=1}^{p} \left(\frac{x_i^2}{4\,000}\right) - \prod_{i=1}^{p} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$x_i \in [-600,\ 600]^3$	0	1.00E-01
Ackley f_5	$f(X) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{p}\sum_{i=1}^{p} x_i^2}\right) - \exp\left(\frac{1}{p}\sum_{i=1}^{p} \cos(2\pi x_i)\right)$	$x_i \in [-30,\ 30]^3$	0	1.00E-05
Schwefel f_6	$f(X) = 418.9829p + \sum_{i=1}^{p} x_i \sin(\sqrt{\lvert x_i \rvert})$	$x_i \in [-512.03,\ 511.97]^2$	0	1.00E-04
Schaffers f_7	$f(X) = 0.5 - \left(\sin^2\sqrt{\sum_{i=1}^{p} x_i^2} - 0.5\right) \Big/ \left[1 + 0.001\sum_{i=1}^{p} x_i^2\right]^2$	$x_i \in [-10,\ 10]^2$	1	1.00E-05

表 2　实验中对比算法的参数设置

Algorithm	Inertia weight			Learning factors		Other parameters		
	w	$w_{\min}$	$w_{\max}$	C_1	C_2			
LWPSO		0	1	2	2			
DPSO	0.6			2	2	$a = 0.99$	$b = 0.99$	$H = 10$
FAPSO	0.5			2	2	$w_0 = 0.5$	$k_1 = 1.5$	$k_2 = 0.3$
IPSO				1.43	1.43	$\delta = 1.0$E-7	$\lambda = 0.002$	$p_m = 0.5$
LSPSO		0	1	2	2	$\delta = 1.0$E-6	$N = 5$	$p_m = 0.5$

表 3 7个标准函数的测试结果对比

Function	Algorithm	Fitness				Iterations			Success rate
		Mean	Best	Worst	Standard deviation	Mean	Min	Max	
f_1	LWPSO	2.120E-37	1.042E-64	5.925E-36	1.082E-36	308	296	320	1
	DPSO	2.218E-43	6.418E-50	3.308E-42	6.770E-43	222	196	251	1
	FAPSO	1.150E-62	9.304E-69	2.549E-61	4.680E-62	166	149	185	1
	IPSO	1.264E-03	1.676E-06	1.920E-02	3.500E-03	500	500	500	0
	LSPSO	8.117E-37	7.111E-65	2.427E-35	4.430E-36	311	291	328	1
f_2	LWPSO	6.668E+04	0.000E+00	1.000E+06	2.537E+05	146	29	500	0.93
	DPSO	3.367E+04	0.000E+00	1.000E+06	1.825E+05	63	7	500	0.93
	FAPSO	3.440E+02	0.000E+00	1.010E+04	1.842E+03	37	4	500	0.97
	IPSO	2.017E+01	1.875E-02	9.832E+01	2.664E+01	67	4	247	1
	LSPSO	3.508E+01	3.707E-15	9.832E+01	4.421E+01	132	2	500	1
f_3	LWPSO	1.327E+00	0.000E+00	4.975E+00	1.023E+00	207	124	500	0.93
	DPSO	1.890E+00	0.000E+00	8.955E+00	2.063E+00	137	7	500	0.83
	FAPSO	2.909E+00	0.000E+00	1.393E+01	3.241E+00	227	29	500	0.67
	IPSO	1.594E+00	6.127E-07	3.980E+00	1.031E+00	239	7	500	0.77
	LSPSO	1.296E+00	3.553E-15	2.985E+00	7.896E-01	237	119	500	0.93
f_4	LWPSO	5.152E-02	0.000E+00	1.774E-01	3.600E-02	191	114	500	0.93
	DPSO	6.482E-02	0.000E+00	2.662E-01	5.500E-02	98	20	500	0.93
	FAPSO	5.068E-02	0.000E+00	2.267E-01	5.060E-02	108	4	500	0.87
	IPSO	7.877E-02	7.396E-03	1.998E-01	4.970E-02	262.47	16	500	0.67
	LSPSO	4.488E-02	7.396E-03	1.651E-01	3.530E-02	206	109	500	0.93
f_5	LWPSO	2.961E-16	-8.882E-16	2.665E-15	1.703E-15	241	219	263	1
	DPSO	1.776E-16	-8.882E-16	2.665E-15	1.656E-15	113	94	145	1
	FAPSO	3.813E-01	-8.882E-16	2.958E+00	8.783E-01	164	52	500	0.83
	IPSO	6.839E-06	7.365E-07	1.246E-05	3.574E-06	289	87	500	0.73
	LSPSO	5.453E-10	4.254E-13	8.572E-09	1.795E-09	242	217	259	1
f_6	LWPSO	1.150E+02	2.546E-05	2.301E+02	6.756E+01	441	119	500	0.17
	DPSO	9.982E+01	2.546E-05	2.301E+02	8.404E+01	349	37	500	0.33
	FAPSO	7.681E+01	2.546E-05	2.301E+02	6.297E+01	331	30	500	0.37
	IPSO	3.839E+00	2.546E-05	1.150E+02	2.100E+01	184	33	500	0.93
	LSPSO	4.218E+01	2.546E-05	1.150E+02	5.639E+01	352	149	500	0.57
f_7	LWPSO	9.984E-01	1.000E+00	9.903E-01	3.700E-03	255	151	500	0.83
	DPSO	9.926E-01	1.000E+00	9.903E-01	4.200E-03	425	37	500	0.23
	FAPSO	9.926E-01	1.000E+00	9.903E-01	4.200E-03	404	39	500	0.23
	IPSO	9.948E-01	1.000E+00	9.903E-01	4.900E-03	355	17	500	0.43
	LSPSO	9.978E-01	1.000E+00	9.903E-01	4.000E-03	275	32	500	0.73

标看: LWPSO在 f_1, f_3, f_4, f_5 和 f_7 中收敛精度较高, 尤其在 f_7 中具有明显优势, 但在 f_2 和 f_6 中性能最差; DPSO在 f_7 中收敛精度低于IPSO, LSPSO和LWPSO, 而在其他函数中与LWPSO性能接近; FAPSO在 f_1 中收敛精度最高, 在 f_2, f_3 和 f_4 中性能也较好, 但在 f_5, f_6 和 f_7 中收敛精度较低, 尤其在 f_5 中明显劣于其他算法; IPSO在 f_2 和 f_6 中收敛精度明显高于其他算法, 但在 f_1 和 f_4 中是各种算法中最差的; 本文给出的LSPSO在 f_2, f_3, f_4 中均得到接近最优的收敛精度, 尤其在 f_1 中明显优于IPSO, 在 f_5 中明显优于FAPSO和IPSO, 在 f_6 中明显优于LWPSO, DPSO和FAPSO, 在 f_7 中明显优于FAPSO, DPSO和IPSO.

LWPSO, DPSO, FAPSO和IPSO在不同函数中性能优劣相差悬殊, 而LSPSO在各实验函数中均具有较好的收敛精度, 在摆脱局部极值和收敛精度等方面具有显著优势, 这说明LSPSO具有较好的寻优性能.

3.2.2 成功率和算法收敛速度

表3同时给出了各种算法30次独立运行达到表1指定收敛精度的迭代次数(均值、最小和最大)和搜索成功率等指标. 若算法在预先设置的最大迭代次数内达到指定收敛精度, 则认为此次运行搜索成功; 否则, 认为搜索失败. 其中

$$\text{成功率(Success rate)} = \text{搜索成功的运行次数/总实验次数}. \tag{16}$$

从实验结果看, LSPSO在测试函数 f_1, f_2, f_3, f_4 和 f_5 中都具有最高的成功率; 在函数 f_6 中IPSO的成功率高于LSPSO, 但在 f_1, f_3, f_4, f_5 和 f_7 中, 其成功率

明显低于 LSPSO; LWPSO 在函数 f_7 中成功率高于 LSPSO, 但在 f_6 中, 其命中率只有 0.17, 远低于 LSPSO 的命中率 (0.57). 从 7 个测试函数整体搜索成功率看, 5 种算法中 LSPSO 较其余 4 种方法具有明显优势, 能以较大概率跳出局部最优, 成功收敛到高精度解.

实验结果表明, 对于较难的复杂优化问题, LSPSO 在收敛精度、成功率和收敛速度等方面均达到了较好的优化效果, 综合指标优于参与对比实验的其他方法. 以上实验结果充分验证了 LSPSO 算法的正确性和高效性.

4 结　论

本文针对标准 PSO 算法的早熟收敛提出了基于进化停滞周期的局部变异粒子群优化算法 (LSPSO), 并从理论上证明了其收敛性. LSPSO 算法以低计算复杂度的种群进化停滞周期判断代替高计算复杂度的多样性计算, 在种群进化停滞时对随机选中的部分粒子执行变异操作, 扩大了种群的多样性和搜索空间; LSPSO 算法以近期全局最优位置作为粒子飞行时的“社会”影响, 避免了因单一全局最优位置对粒子约束引起的早熟收敛. 对 7 个经典测试基准函数进行优化实验, 结果表明, LSPSO 算法能够极大地提高收敛速度和精度, 能够有效摆脱局部极值点, 提高寻优成功率.

参考文献(References)

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

[2] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of the 6th Int Symposium on Micro Machine and Human Science. Nagoya: IEEE Service Center, 1995: 39-43.

[3] 薛文涛, 吴晓蓓, 徐志良. 用于多峰函数优化的免疫粒子群网络算法[J]. 系统工程与电子技术, 2009, 31(3): 705-709.
(Xue W T, Wu X B, Xu Z L. Immune particle swarm network algorithm for multimodal function optimization[J]. Systems Engineering and Electronics, 2009, 31(3): 705-709.)

[4] 高亮, 杨林, 周驰, 等. 基于粒子群优化的神经网络训练算法在产品种类预测中的应用[J]. 计算机集成制造系统, 2006, 12(3): 465-469.
(Gao L, Yang L, Zhou C, et al. Category forecast application of neural network algorithm trained by particle swarm optimization[J]. Computer Integrated Manufacturing Systems, 2006, 12(3): 465-469.)

[5] Fourie P C, Groenwold A A. The particle swarm optimization algorithm in size and shape optimization[J]. Structural and Multidisciplinary Optimization, 2002, 23(4): 259-267.

[6] Ali A, Al-Anzi F S. A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application[J]. Computers and Operations Research, 2006, 33(4): 1056-1080.

[7] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.

[8] 方伟, 孙俊, 须文波. 一种多样性控制的粒子群优化算法[J]. 控制与决策, 2008, 23(8): 863-868.
(Fang W, Sun J, Xu W B. Diversity-controlled particle swarm optimization algorithm[J]. Control and Decision, 2008, 23(8): 863-868.)

[9] 巩敦卫, 张勇, 张建化, 等. 新型粒子群优化算法[J]. 控制理论与应用, 2008, 25(1): 111-114.
(Gong D W, Zhang Y, Zhang J H, et al. Novel particle swarm optimization algorithm[J]. Control Theory and Applications, 2008, 25(1): 111-114.)

[10] 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.)

[11] 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]. OR Trans, 1999, 3(4): 79-83.)

[12] 韩江洪, 李正荣, 魏振春. 一种自适应粒子群优化算法及其仿真研究[J]. 系统仿真学报, 2006, 18(10): 2969-2971.
(Han J H, Li Z R, Wei Z C. Adaptive particle swarm optimization algorithm and simulation[J]. J of System Simulation, 2006, 18(10): 2969- 2971.)

[13] 罗辞勇, 陈民铀. 克服恋食行为的 PSO 算法改进研究[J]. 控制与决策, 2008, 23(7): 776-780.
(Luo C Y, Chen M Y. Improved PSO algorithm with overcoming behaviour of indulging in food[J]. Control and Decision, 2008, 23(7): 776-780.)