

文章编号: 1001-0920(2009)06-0942-04

一类新的粒子群算法

孟学雷^{1,2}, 贾利民¹

(1. 北京交通大学 轨道交通控制与安全国家重点实验室, 北京 100044; 2. 兰州交通大学 交通运输学院, 兰州 730070)

摘要: 提出一种新的模糊粒子群优化算法)) 收敛模糊粒子群优化算法. 重点研究了收敛因子的确定和模糊隶属度函数的选择对算法性能的影响. 在考虑计算效率的同时, 提高了算法的精度. 利用 4 个基准函数测试了收敛模糊粒子群优化算法的性能, 并与模糊粒子群优化算法、收敛粒子群优化算法以及基本粒子群优化算法进行了对比. 实验结果表明, 新算法具有很好的性能.

关键词: 粒子群; 收敛; 模糊

中图分类号: TP18 文献标识码: A

A new kind of particle swarm algorithm

MENG Xuelei^{1,2}, JIA Li2min¹

(1. State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China; 2. School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China. Correspondent: MENG Xuelei, E2mail: foxred@163.com)

Abstract: A new kind of fuzzy particle swarm optimization (FPSO) algorithm is proposed, called convergent fuzzy particle swarm optimization (CFPSO) algorithm. It focuses on the impact on the algorithm performance of the convergent gene computation and membership function selection, which both improves the precision and increases the efficiency. It tests the performance of the new algorithm with four benchmark functions and is compared with FPSO, CPSO (convergent PSO) and PSO. Experiment results show that CFPSO has excellent performance.

Key words: Particle swarm optimization (PSO); Convergent; Fuzzy

1 引言

粒子群优化 (PSO)^[1] 算法是一种进化计算技术, 最早是由 Kennedy 和 Eberhart 于 1995 年提出的. 该算法源于对鸟群捕食行为的研究, 是一种基于迭代技术的优化工具, 用来解决非线性、多极值的复杂优化问题. 这种优化方法简单易懂, 易于建模实现^[2,3], 自提出以来, 得到了学术界的广泛关注. 文献 [4] 提出了模糊粒子群优化算法, 但对粒子运动方程中隶属度函数的选择并没有作深入探讨; [5] 提出了基于本能的满意度粒子群优化算法; [6] 对粒子群优化算法作了改进; [7] 对模糊适应粒子群优化算法进行了研究; [8] 研究了利用模糊适应粒子群优化算法解决单位委托计算问题; [9, 10] 研究了利用混合粒子群优化算法进行模糊规则的生成及隶属度函数的选择问题; [11] 等研究了基于粒子群优化算法的加权模糊推理问题. 然而, 上述文献都没有通过实验对

算法的效率和精度进行综合研究.

本文对模糊粒子群优化算法作了进一步的改进, 将收敛因子引入粒子的运动方程中, 形成了新的粒子群优化算法)) 收敛模糊粒子群优化算法. 对隶属度函数的选择以及收敛因子的确定进行了深入的研究, 综合考虑了算法的效率和精度.

2 模糊粒子群优化算法

2.1 基本粒子群优化算法

粒子群优化算法是受自然界中鸟群、鱼群觅食过程中出现的群体智能现象启发而提出的^[1]. 假设 i 是一个解, N 表示 i 解向量中元素的个数. 每个粒子包含两个 N 维向量: 一个是布尔向量 x_i , 代表一个候选解, 或称为 i 的状态; 另一个是实数向量, 称为粒子的速度. 在粒子群算法中, 速度向量代表了粒子以什么速度、朝什么方向、在解的每个维度上如何飞行.

收稿日期: 2008-05-10; 修回日期: 2008-07-31.

基金项目: 国家自然科学基金重点项目 (600332020); 国家自然科学基金项目 (60674001).

作者简介: 孟学雷 (1979), 男, 山东泰安人, 讲师, 博士, 从事交通运输系统智能优化技术的研究; 贾利民 (1963), 男, 新疆阿勒泰人, 教授, 博士生导师, 从事智能控制及智能自动化理论与应用等研究.

令 $K(i)$ 表示粒子 i 的邻居, 用 p_i 表示粒子 i 所经过的最优位置. 在每次迭代过程中, 每个粒子 i 根据下列方程调整其状态和飞行速度:

$$v_{i+1} = Xv_i + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i), \quad (1)$$

$$x_{i+1} = x_i + v_{i+1}. \quad (2)$$

其中: X 称为惯性因子, 取值介于 $[0, 1]$, 通常随着迭代的进行而逐渐减小^[12]; c_1 和 c_2 是两个常量, 通常情况下取 $c_1 + c_2 = 4$ ^[13], 表示粒子群体对单个粒子的影响程度; r_1 和 r_2 是介于 0 和 1 之间的两个随机数; p_g 代表粒子群体中所有粒子所经历过的最优位置.

2.2 模糊粒子群优化算法

模糊粒子群优化(FPSO)算法^[4]是基本粒子群优化算法的泛化. 它与基本粒子群优化算法的不同在于, 并不只是最优粒子能够影响其他相邻粒子, 而是若干粒子均可以影响其周围相邻的其他粒子. 每个粒子受周围其他相邻粒子影响的程度取决于这些粒子影响作用的隶属度, 而吸引力的大小则取决于粒子相对应的模糊变量. 这样便充分利用了粒子周围各个粒子的优点, 使算法具有较高的精度. 但是, 文献[4]并没有对隶属度函数的选择以及不同的隶属度函数对算法性能造成的影响作深入探讨.

隶属度是一个模糊变量. Triangle, Trapezoidal, Gaussian, Bell 和 Sigmoid 函数是几种常见的隶属度函数(MF)^[14].

人为确定一个常数 k , 代表某一粒子周围存在的能对该粒子的运动产生影响的邻居粒子个数. 根据问题求解的精度要求 k 可以变化. 令 h 是 k 个最优粒子中的一个, $f(p_g)$ 表示最优粒子的适应度值.

如果隶属度函数是基于 Bell 函数建立的, 则对于粒子群优化算法, 隶属度函数可定义为

$$U(h) = \frac{1}{1 + \left(\frac{f(p_h) - f(p_g)}{B}\right)^2}. \quad (3)$$

对于粒子群优化算法而言, 如果隶属度函数是基于 Gaussian 函数建立的, 则其隶属度函数定义为

$$U(h) = \exp\left(-\frac{1}{2} \left(\frac{f(p_h) - f(p_g)}{B}\right)^2\right). \quad (4)$$

如果隶属度函数是基于 Sigmoid 函数建立的, 则其隶属度函数定义为

$$U(h) = \frac{1}{1 + \exp(-B(f(p_h) - f(p_g)))}. \quad (5)$$

如果隶属度函数是基于 Triangle 函数建立的, 则其隶属度函数定义为

$$U(h) = \max\left(\min\left(B \frac{f(p_h) - f(p_g^0)}{f(p_g) - f(p_g^0)}, 1\right), 0\right).$$

$$B \frac{f(p_g^1) - f(p_h)}{f(p_g^1) - f(p_g)}, 0). \quad (6)$$

如果隶属度函数是基于 Trapezoid 函数建立的, 则其隶属度函数定义为

$$U(h) = \max\left(\min\left(B \frac{f(p_h) - f(p_g^0)}{f(p_g) - f(p_g^0)}, 1\right), B \frac{f(p_g^1) - f(p_h)}{f(p_g^1) - f(p_g)}, 0\right). \quad (7)$$

因为 $f(p_h) \leq f(p_g)$, 所以 $U(h)$ 是一个递减函数. 当 $f(p_h) = f(p_g)$ 时, 函数值为 1. 当 $f(p_h)$ 与 $f(p_g)$ 之间的差距越来越大时, 函数值趋近于 0. 为了避免适应度函数带来过多的影响, 定义 $B = f(p_g)/l$, 其中 l 是用户给定的参数. 对于一个确定的 $f(p_h)$, l 越大, $U(h)$ 越小. $f(p_g^0)$ 与 $f(p_g^1)$ 分别是 Triangle 和 Trapezoidal 两个函数的两个边界值. 在模糊粒子群优化算法中, 速度方程定义为^[4]

$$v_{i+1} = Xv_i + c_1 r_1 (p_i - x_i) + \sum_{h \in B(i, k)} U(h) c_2 r_2 (p_h - x_i), \quad (8)$$

其中 $B(i, k)$ 表示与粒子 i 相邻的 k 个粒子的集合. 每个粒子受其自身最优位置 p_i 和其周围这 k 个最优粒子的影响, 而每个相邻粒子对其影响的程度取决于 $U(h)$ 的值. 可以推断, 当 k 值取为 1 时, 模糊粒子群优化算法将蜕变为基本粒子群优化算法.

3 收敛模糊粒子群优化算法

3.1 收敛粒子群优化算法

Clerc 在文献[2]中严格证明了引入收敛因子后粒子群优化算法一定是收敛的. 在粒子运动方程中, c_1 和 c_2 是两个常数, 通常在基本粒子群优化算法中设置为 2. 在 Clerc 的模型中, U 定义为 c_1, c_2 的和. 收敛因子 k 可定义为

$$k = 2 / |2 - U - \sqrt{|U^2 - 4U|}|. \quad (9)$$

于是, 收敛粒子群优化(CPSO)算法的运动方程为

$$v_{i+1} = k [Xv_i + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i)]. \quad (10)$$

该模型可以视为基本粒子群优化算法的泛化. 当 $U = 4$ 时, $k = 1$, 则该算法退化为基本粒子群优化算法, 难以收敛. 根据 Clerc 的结论^[2], 当 $U > 4$ 时, 算法具有很强的收敛特性, 其收敛特性几近于线性. 当 $U = 4$ 时, 算法易螺旋收敛于非平凡解.

3.2 收敛模糊粒子群优化算法

基本粒子群优化算法的缺点是难以收敛. 引入收敛因子之后, 可保证其较快的收敛速度^[15, 16], 但算法的精确度还有待于提高. 因此, 本文将模糊粒子群优化算法中隶属度函数的概念引入模型, 提出一种新的粒子群优化算法))) 收敛模糊粒子群优化(CFPSO)算法. 该算法与模糊粒子群优化算法的区

别是, 在粒子的运动方程中加入了收敛因子. 其运动方程为

$$v_{i+1} = k_1 Xv_i + c_1 r_1 (p_i - x_i) + \sum_{h=1}^H U(h) c_2 r_2 (p_h - x_i). \quad (11)$$

这样, 粒子群优化算法的粒子运动方程中既引入了收敛因子, 可保证算法的收敛性能; 同时又引入了模糊算子, 提高了算法的精度.

4 参数设置与计算结果分析

4.1 收敛因子确定

根据 Clerc 的模型, $U = c_1 + c_2$, 收敛因子 k 定义为 $k = 2 / |2 - U - \sqrt{|U - 4U|}|$. 于是 c_1 和 c_2 的取值便成为问题的关键. 为了确定 c_1 和 c_2 的值, 使算法具有良好的性能, 采用 Rastrigrin 函数(见 4.3 节) 作为测试函数进行数值计算实验. 粒子群规模设计为 100, 每个粒子有 20 维. 循环迭代 100 次.

图 1 和表 1 分别给出了目标函数值与 c_1 和 c_2 的关系.

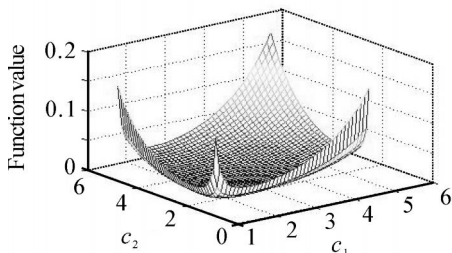


图 1 目标函数值与 c_1, c_2 关系曲面

表 1 目标函数值与 c_1, c_2 关系

c_1	c_2			
	1.9	2.0	2.1	2.2
1.8	1.342E205	9.024E06	2.021E206	2.629E205
1.9	1.083E205	9.048E06	1.021E206	2.124E206
2.0	8.461E206	1.049E05	0.041E206	1.027E206
2.1	1.257E205	1.143E05	1.083E206	6.145E206
2.2	1.062E205	1.029E05	1.672E206	1.029E205

从图 1 可以看出, 随着 c_1 和 c_2 分别各自接近 2.0 和 2.1, 目标函数的值越来越小. 通过表 1 可以推断, 当 c_1 和 c_2 分别取 2.0 和 2.1 时, 目标函数的值最小. 又根据 Clerc 的 $U > 4$ 时算法具有较好的收敛性能的结论, 此时 $U = 2.0 + 2.1 = 4.1 > 4$, 所以取 $c_1 = 2.0, c_2 = 2.1$.

4.2 隶属度函数选择

为了从 Triangle, Trapezoidal, Gaussian, Sigmoid 和 Bell 这 5 个常用的模糊隶属度函数中为粒子群运动方程选择适宜的函数, 以 f_1 作为标准函数, 分别采用 5 个函数作为隶属度函数, 各运行 100 次. 粒子群的规模是 100, 每个粒子含有 20 维, 标准

函数的目标值是 $1.0000E207$, 粒子的最大速度 V_{max} 限定为 10.

表 2 中列出了运用各种隶属度函数计算出的标准函数的最优解、平均解和达到目标值的次数. 运用三角隶属度函数, 可以 30 次达到目标值, 比运用其他 4 种函数的次数都少. 说明三角隶属度函数虽然计算效率高, 但其计算精度是 5 个模糊隶属度函数中最低的. 另外可以看到, 运用 Bell 函数所得标准方程的最优解和平均解均优于其他 4 种函数, 并且其达到目标值的次数也最多, 所以说 Bell 函数具有最高精度.

表 2 目标函数值与隶属度函数关系

MF	Best solution	Average solution	Reach times
Gaussian	5.1287E11	3.4687E207	58
Bell	8.2256E11	1.6721E207	55
Sigmoid	6.1259E11	4.2684E207	53
Triangle	2.4511E08	8.5784E204	30
Trapezoid	1.6458E08	7.2645E204	31

图 2 给出了运用不同隶属度函数进行计算时, 达到最优值的次数与消耗时间的关系. 由图可知, 若要达到相同次数的目标函数值, 三角函数所耗费的时间最少. 虽然三角函数在 5 种函数中计算精度最低, 但效率却最高. 这是因为三角函数的计算复杂度要远低于其他几种函数. 因此, 可通过更换粒子运动方程中的隶属度函数, 在计算效率和计算精度之间寻求平衡.

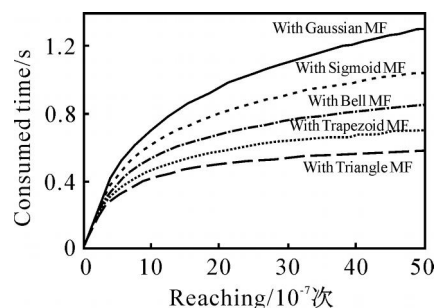


图 2 隶属度函数与计算性能之间的关系

4.3 计算结果分析

本文采用学术界常用的 4 个标准函数作为测试函数, 运行 100 次, 对新算法的性能进行分析. 粒子群规模是 100. 测试时, 粒子的维数分别取 20, 40 和 60.

f_1 函数是泛化的 Rastrigrin 函数, 即

$$f_1(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad x_i \in [-5.12, 5.12]; \quad (12)$$

f_2 函数是泛化的 Griewank 函数, 即

$$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad x_i \in [-600, 600]; \quad (13)$$

f_3 函数是泛化的 Rosenbrock 函数, 即

$$f_3(x) = \sum_{i=1}^n (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2), \quad x_i \in [-30, 30]; \quad (14)$$

f_4 函数是泛化的 Ackley 函数, 即

$$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e, \quad x_i \in [-30, 30]. \quad (15)$$

表 3 列出了分别使用 4 种算法计算得到标准函数 $f_1 \sim f_4$ 的结果. 使用 FPSO 算法计算所得的 4 个函数的最优解值和平均解值是 4 种算法中最小的, 所以 FPSO 算法的精度最高. 该实验在运行 100 次的情况下, 运用 CFPSO 算法所达到目标值的次数多于采用 PSO 和 CPSO, 但略少于采用 FPSO 算法计算所得到的次数.

表 3 4 个标准函数实验数据

Function	Algorithm	Best solution	Average solution	Reach times
f_1	PSO	2.472E11	3.081E202	27
	FPSO	1.421E11	8.224E203	54
	CPSO	2.472E11	3.045E202	27
	CFPSO	1.713E11	9.441E203	49
f_2	PSO	2.341E14	9.384E208	77
	FPSO	5.672E15	8.243E208	82
	CPSO	2.436E14	2.069E207	78
	CFPSO	7.882E15	9.561E208	79
f_3	PSO	4.349E11	8.796E206	73
	FPSO	6.821E12	3.710E206	79
	CPSO	8.341E12	8.421E206	74
	CFPSO	7.610E12	6.109E206	78
f_4	PSO	7.120E11	2.201E204	50
	FPSO	6.421E12	2.360E205	66
	CPSO	3.246E11	5.560E205	50
	CFPSO	9.040E12	3.264E205	63

如图 3 所示, 随着达到目标值次数的增加, 所消耗的计算时间也随之增加. FPSO 算法消耗的时间与基本 PSO 算法所消耗的时间很接近, 所以选取三角函数作为模糊隶属度函数, 对算法的计算效率影响较小. 而 CFPSO 算法所消耗的时间比 FPSO 算法

少, 是因为算法中收敛因子起了作用. 另外还可看到, 达到目标值的次数较少时, CFPSO 算法耗时反而比基本 PSO 算法所消耗的时间多, 这是因为在计算收敛因子数值时, 消耗了一部分时间.

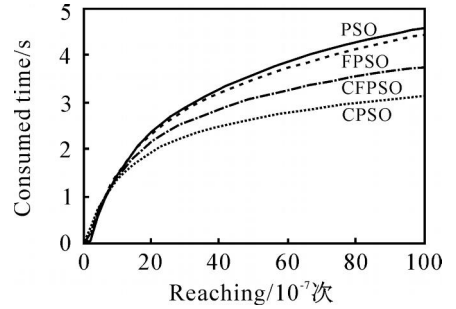


图 3 算法的计算性能对比

由分析可知, CPSO 算法的计算效率最高, FPSO 算法的计算精度最高. 而 CFPSO 算法继承了两者的优点, 既考虑了算法的精度, 又考虑了效率, 因此具有最好的算法性能.

5 结 论

本文提出了一种新的粒子群优化算法))) 收敛模糊粒子群优化算法. 该算法秉承了模糊粒子群优化算法高精度以及收敛粒子群优化算法高效率的特点, 具有很好的计算性能. 通过改变收敛模糊粒子群优化算法速度公式中的模糊隶属度函数, 可以在效率与精度之间寻求平衡.

今后研究的重点是将该算法应用于各工程领域, 发挥其强大的计算功能.

参考文献(References)

- [1] Eberhat R, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of 6th Int Symposium on Micro Machine and Human Science. Nagoya, 1995: 392-43.
- [2] Clerc M, Kennedy J. The particle swarm explosion, stability, and convergence in multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [3] Kennedy J, Eberhart R C. Swarm intelligence[M]. San Francisco: Morgan Kaufmann, 2001.
- [4] Ashraf M Abdelbar, Suzan Abdelshahid, Donald C Wunsch II. Fuzzy PSO: A generation of particle swarm optimization[C]. Proc of Int Joint Conf on Neural Networks. Montreal, 2005: 108621091.
- [5] Abdelbar A M, Abdelshahid S. Instinct2based PSO with local search applied to satisfiability[C]. Proc of Int Joint Conf on Neural Networks. Budapest, 2004, 3: 22912-2295.

控制器的设计方法. 文中考虑的控制函数可以是足够光滑的奇函数, 它包含了通常的线性函数. 另外, 例如函数 $U(u) = u + u^2$ 也满足定理 1 的条件, 仍可利用本文定理 2 的方法考虑 H_1 控制器设计. 因此对此类系统的控制问题研究具有很好的实际意义.

参考文献(References)

- [1] Zames G. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminars and approximation inverses[J]. IEEE Trans on Automatic Control, 1981, 26(1): 30-32.
- [2] Iwasaki T, Skelton R E. All controllers for the general H_1 control problem: LMI existence conditions and state space formulas[J]. Automatica, 1994, 30(8): 1307-1317.
- [3] Apkarian P, Gahinet P. A convex characterization of gain-scheduled H_1 controller [J]. IEEE Trans on Automatic Control, 1995, 40(5): 853-864.
- [4] Lee Y S, Moon Y S, Kwon W H, et al. Delay-dependent robust H_1 control for uncertain systems with a state-delay[J]. Automatica, 2004, 40(1): 65-72.
- [5] Wu M, He Y, She J H. Delay-dependent criteria for robust stability of time-varying delay systems [J]. Automatica, 2004, 40(8): 1432-1439.
- [6] Jiang X F, Han Q L. On H_1 control for linear systems with interval time-varying delay[J]. Automatica, 2005, 41(12): 2099-2106.
- [7] Xu S Y, James L, Zou Y. New results on delay-dependent robust H_1 control for systems with time-varying delays[J]. Automatica, 2006, 42(2): 343-348.
- [8] Chen B, Liu X P. Delay-dependent robust H_1 control for T2S fuzzy systems with time delay[J]. IEEE Trans on fuzzy system, 2005, 13(4): 542-556.
- [9] Xu S Y, James L. Improved delay-dependent stability criteria for time-delay systems [J]. IEEE Trans on Automatic Control, 2005, 50(3): 384-387.
- [10] 梅生伟, 申铁龙, 刘志康. 现代鲁棒控制理论与应用[M]. 北京: 清华大学出版社, 2003.
(Mei S W, Shen T L, Liu Z K. Modern robust control theory and application [M]. Beijing: Tsinghua University Press, 2003.)
- [11] 哈里尔. 非线性系统[M]. 第 3 版. 北京: 电子工业出版社, 2005.
(Khalil H K. Nonlinear systems[M]. 3rd ed. Beijing: Publishing House of Electronics Industry, 2005.)
- [12] Xie L. Output feedback H_1 control of systems with parameter uncertainty[J]. Int J of Control, 1999, 63(4): 742-750.

(上接第 944 页)

- [6] Yang Guangyou. A modified particle swarm optimizer algorithm [C]. The 8th Int Conf on Electronic Measurement and Instruments. Xi an, 2007, 2: 6752-679.
- [7] Bajpai P, Singh. Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market[J]. IEEE Trans on Power Systems, 2007, 22(4): 2152-2160.
- [8] Saber A Y, Senjyu T, Yona A, et al. Unit commitment computation by fuzzy adaptive particle swarm optimization [J]. IET Generation Transmission and Distribution, 2007, 1(3): 456-465.
- [9] Esmine A A A. Generating fuzzy rules from examples using the particle swarm optimization algorithm[C]. 7th Int Conf on Hybrid Intelligent Systems. Kaiserslautern, 2006: 340-343.
- [10] Esmine A A A, Lamber-Torres A G. Fitting fuzzy membership functions using hybrid particle swarm optimization[C]. Proc of IEEE Int Conf on Fuzzy System. Vancouver, 2006: 2112-2119.
- [11] Sufang An, Kunqi Liu, Bo Liu. Improved weighted fuzzy reasoning algorithm based on particle swarm optimization[C]. Proc of 6th Int Conf on Machine Learning and Cybernetics. Hong Kong, 2007: 1304-1308.
- [12] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]. Proc of IEEE Int Conf on Evolutionary Computation. Alaska, 1998: 62-73.
- [13] Murata Hideki, Aiyoshi Eitaro. Bifurcation and convergence of particle swarm optimization dynamics embedded into upper and lower bound[J]. IEEE Trans on Electronics, Information and Systems, 2006, 126(7): 904-912.
- [14] Jang J S R, Sun C T, Mizutani E. Neuro-fuzzy and soft computing[M]. New Jersey: Prentice-Hall, 1996.
- [15] van den Bergh F, Engelbrecht A P. A new locally convergent particle swarm optimizer[C]. Proc of IEEE Int Conf on Systems, Man and Cybernetics. Yasmine Hammamet, 2002: 94-99.
- [16] Trelea I C. The particle swarm optimization algorithm: Convergence analysis and parameter selection [J]. Information Processing Letters, 2003, 85(6): 317-325.