第 21 卷 第 3 期 控 制 与 决 策 2006 年 3 月
Vol. 21 No. 3 Control and Decision Mar. 2006

文章编号: 1001-0920(2006)03-0263-04

基于混沌序列的粒子群优化算法

孟红记, 郑 鹏, 梅国晖, 谢 植
(东北大学 信息科学与工程学院, 沈阳 110004)

摘 要: 提出一种改进粒子群局部搜索能力的优化算法. 对于陷入局部极小点的惰性粒子, 引入混沌序列重新初始化, 在迭代中产生局部最优解的邻域点, 帮助惰性粒子逃离束缚并快速搜寻到最优解. 对经典函数的测试计算表明, 改进的混合算法通过微粒自适应更新机制确保了全局搜索性能和局部搜索性能的动态平衡, 而且保持了 PSO 计算简洁的特点, 在收敛速度和精度上均优于普通的 PSO 算法.

关键词: 粒子群; 混沌序列; 优化; 局部极小点
中图分类号: TP18 **文献标识码**: A

Particle Swarm Optimization Algorithm Based on Chaotic Series

MENG Hong-ji, ZHENG Peng, MEI Guo-hui, XIE Zhi
(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: MENG Hong-ji, E-mail: meng010729@sina.com)

Abstract: An advanced particle swarm optimization algorithm is presented to enhance the local searching ability. Some particles trapped in local minimums are initialized again by chaotic series in order to introduce neighboring regions of local minimums in the iteration and help them break away from local optimum to find the globe optimal solution rapidly. The experimental results of classic functions show that the improved hybrid method makes use of the ergodicity of chaotic search to improve the capability of precise search and keep the balance between the global search and the local search, and maintain the concise calculation of particle swarm optimization (PSO) property. The enhanced algorithm has great advantage of convergence property and robustness compared to genetic algorithm and PSO algorithm.

Key words: Particle swarm; Chaotic series; Optimization; Local minimums

1 引 言

粒子群优化算法(Particle swarm optimization, PSO)是由 Kennedy 等于 1995 年提出的一种模拟鸟类捕食行为的全局优化算法, 通过鸟群之间的集体协作来达到目的[1]. 与其他生物演化算法类似, PSO 算法是一种基于迭代过程的优化方法. 该方法首先由系统初始化一组随机解; 然后通过迭代搜寻最优值.

PSO 算法由于只涉及初等运算, 具有计算简洁、易于实现、需要调整的参数少等特点, 近年来取得了较快发展. 目前在各类多维连续空间优化问题、神经网络训练、模糊系统控制、组合优化和机器人路径规划等领域均取得了非常好的效果[2]. 与其他全局优化方法一样, 粒子群算法同样存在易于陷入局部极值束缚的特点, 尤其对于比较复杂的多峰搜索问题. 目前学者们提出了许多改进算法来加强局部搜索能力, 但仍存在一些问题. 如免疫 PSO 算法[3]和Multistart PSO 算法[4]等改进算法, 尽管摆脱了局部最优点的束缚, 但由于粒子的重新初始化而破坏了当前粒子的结构, 降低了收敛速度. Angeline 提出的选择 PSO 算法虽然提高了局部搜索能力, 但同时也削弱了全局搜索能力; 而 Shi 提出的模糊惯

收稿日期: 2005-02-23; **修回日期**: 2005-04-15.
基金项目: 国家自然科学基金项目(50174021).
作者简介: 孟红记(1970—), 女, 河南睢县人, 博士生, 从事过程控制及优化的研究; 谢植(1957—), 男, 沈阳人, 教授, 博士生导师, 从事工业过程参数检测、过程控制及优化的研究.

性权重 PSO 算法则存在不便使用的特点[5,6].

本文提出一种将混沌序列引入粒子群优化的算法. 对于陷入局部极小点的惰性粒子, 引入混沌序列重新初始化, 迭代产生局部最优解的邻域点, 帮助惰性粒子逃离束缚并快速搜寻到最优解. 由于该算法仅对个体粒子重新初始化, PSO 算法迭代中粒子群的当前结构没有改变, 改进算法保持了 PSO 运算和结构简单的特点. 对单峰和多峰经典函数的测试计算表明, 与遗传算法和粒子群算法相比, 改进算法的收敛性能有显著提高.

2　基本的粒子群优化算法

PSO 算法与其他演化算法相似, 也是基于群体的优化方法, 根据对环境的适应度, 将群体中的个体向好的区域移动. 在 PSO 系统中, 每个备选解被称为一个"粒子"(Particle), 多个粒子共存、合作寻优, 每个粒子根据自身"经验"和群体的最佳"经验"在问题空间中向更好的位置"飞行", 搜索最优解.

在每一次迭代中, 粒子通过跟踪个体极值和全局极值来更新自己. PSO 算法的数学过程表示如下:

假定搜索的空间为 D 维, 令 num 表示粒子总数, 第 i 个微粒位置向量表示为 $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, 速度向量表示为 $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$. 粒子经历过的最好位置记为 $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, 也称为 p_{best}. 在群体中所有微粒经历过的最好位置的索引号用符号 g 表示, 即 $P_g = (p_{g1}, g_{g2}, \cdots, g_{gD})$, 也称为 g_{best}.

粒子在每一次迭代找到上述两个极值后, 对自己的速度和位置进行如下更新:

$$v_{id} = w v_{id} + c_1 \text{rand}_1 (p_{id} - x_{id}) + c_2 \text{rand}_2 (p_{gd} - x_{id}); \tag{1}$$

$$x_{id} = x_{id} + v_{id}. \tag{2}$$

其中: $1 \leq i \leq \text{num}$, $1 \leq d \leq D$; c_1, c_2 称为加速因子, 通常取 $c_1 = c_2 = 2$; rand_1 和 rand_2 为两个在 $[0, 1]$ 范围内变化的随机函数; w 为惯性因子, 研究表明较大的 w 有利于跳出局部极小点, 较小的 w 有利于算法收敛. 随着迭代进行, 速度惯性因子 w 由最大 $w_{\max}$ 线性减小到最小 $w_{\min}$, 即

$$w = w_{\max} - n \frac{(w_{\max} - w_{\min})}{N}. \tag{3}$$

其中: n 为当前迭代次数, N 为总迭代次数, $w_{\max} = 0.9$, $w_{\min} = 0.1$.

3　嵌入混沌序列的混合粒子群算法

分析式(1) 和(2) 可以发现, 当某些微粒的位置及其 p_{best} 接近群体的 g_{best} 时, 其速度更新由 $w v_{id}$ 决定. $w < 1$ 时, 粒子的运行速度越来越小, 接近于零, 粒子运行出现"惰性". 随着演化的进行, 其他粒子将很快聚集到这些惰性粒子的周围, 使演化过早地收敛到局部最优点. 另外, 由于粒子初始化和演化过程的随机性, 使 g_{best} 和 p_{best} 的更新带有一定的盲目性, 影响了演化过程的收敛. 为了使微粒群能够跳出此种停滞状态, 惰性粒子应由一个新的有活力的粒子取代. 因此, 本文提出采用混沌搜索来完成"惰性"粒子的重新初始化. 由于混沌序列具有遍历性的特点[7], 即混沌序列可以在一个特定区域内不重复地历经所有状态, 使其成为一种十分有效的搜索工具. 引入混沌序列的搜索算法可在迭代中产生局部最优解的许多邻域点, 以此帮助惰性粒子逃离局部极小点, 并快速搜寻到最优解.

混沌搜索的主要思路是通过某特定格式迭代产生混沌序列; 然后通过载波的方式将混沌变量的值域"放大"到优化变量的取值范围空间. 这种嵌入混沌序列的混合粒子群优化算法简称为 CPSO (Chaotic particle swarm optimization).

混沌搜索的数学过程如下:

如果有粒子停滞时, 产生一个 D 维随机初始向量 $y_0 = [y_{0,1}, y_{0,2}, \cdots, y_{0,D}]'$, $y_{0,d} \in [0, 1]$, 且各分量值之间有微小差别. 用向量 y_0 作为迭代初始值, 根据 Logistic 方程

$$y_{n+1,d} = \mu y_{n,d}(1 - y_{n,d}), \tag{4}$$

开始混沌序列迭代, 得到迭代序列 $y_{n,d}$. 式中: $n = 0, 1, \cdots, N_{\max}$; $d = 1, 2, \cdots, D$.

此格式可以迭代产生局部最优解周围的多个邻域; 然后通过载波方式, 根据

$$y'_{n,d} = x_{i,d} + R_{i,d}(2y_{n,d} - 1), \tag{5}$$

将混沌迭代变量 $y_{n,d}$ 转化为优化变量 $y'_{n,d}$, 即将混沌变量 $y_{n,d}$ 的取值"放大"到一个以粒子当前位置 $x_{i,d}$ 为中心, 以 $R_{i,d}$ 为半径的区域上. 其中: $R_{i,d}$ 为混沌搜索半径; $y'_{n,d}$ 根据函数变量 $x_{i,d}$ 的初始化范围确定, 取值区间如下:

$$y'_{n,d} \in [x_{i,d} - R_{i,d}, x_{i,d} + R_{i,d}].$$

计算函数适应值 $f(y_n)$, 并更新在混沌迭代过程中的历史最优适应值 f^* 和历史最优位置 x_i^*. 如果 f^* 优于 F_i, 则用位置 x_i^* 和速度 v_i^* 替换此粒子的原有位置和速度. 其中

$$v_i^* = \frac{x_i^* - x_i}{\| x_i^* - x_i \|}. \tag{6}$$

嵌入混沌序列的粒子群算法 CPSO 的流程如下:

Step 1: 设置最大进化代数, 随机初始化每个粒子的位置和速度.

Step 2: 计算每个粒子的适应度 F_i, 更新个体历

史最优适应度 $F_{p_{best}}$ 和全局历史最优适应度 $F_{g_{best}}$.

Step 3: 更新个体历史最优位置和全局历史最优位置.

Step 4: 判定每个粒子是否停滞. 采用如下判定标准:

$$\Delta F_i = (F_i - F_{p_{best}}) / F_i \quad (7)$$

式中: δ为预先设定的常数阈值, N_c 为预先设定的常数. 如果迭代中连续 N_c 次不满足条件 $\Delta F_i < \delta$, 则跳至 Step 5; 否则, 粒子运动停滞, 进入混沌搜索过程:

1) 设置当前迭代计数器 $G_c = 1$ 和最大迭代次数 N_{max};

2) 按式(4) 产生混沌序列, 并按式(5) 将其放大到优化取值空间;

3) 计算函数适应值 $f(y_n)$, 并更新最优 f^* 和 x_i^*;

4) 更新迭代计数器 $G_c = G_c + 1$, 并跳转至步骤 2), 直到 $G_c > N_{max}$;

5) 用位置 x_i^* 和速度 v_i^* 替换此粒子的原有位置和速度.

Step 5: 根据式(1) 和(2) 计算每个粒子的新速度值, 并移动粒子到新的位置上.

Step 6: 如果满足收敛准则, 则停止; 否则, 转 Step 2.

4　改进的粒子群算法性能分析

下面通过 3 个 Benchmark 函数 (Schwefel, Rosenbrock 和 Schaffe) 优化问题测试本文算法的性能[7], 并与遗传算法 SGA, 自适应粒子群算法 APSO[8], 标准粒子群算法 SPSO 测试结果进行比较.

为比较方便, 每个函数的变量初始化范围和最大粒子速度 V_{max} 相应列在表 1 中.

表 1　变量初始化范围

函数	变量范围	最大速度	最小值	维数
f_1	(-500, 500)	60	-12 569.5	30
			-4 189.8	10
f_2	(-30, 30)	5	0	30
				10
f_3	(-100, 100)	15	-1	2

对每个函数随机进行 100 次实验. 对于 CPSO 算法, 阈值 δ取 1e-3, 常数 N_c 取 6, 搜索半径 $R_{i,d}$ 设为函数自变量 x 定义域的 20%.

1) Schwefel 函数

$$f_1(x) = -\sum_{i=1}^{D} (x_i \sin(\sqrt{|x_i|})). \quad (8)$$

在计算中, 最大演化代数 G_{max} 设为 500, 对每个函数取不同维数且不同个体数目 m 的种群进行测试. m 分别取为 15 和 30, 维数分别取 10 和 30.

对 Schwefel 函数的计算结果如表 2 所示. 在相同维数和个体数目条件下, 与其他 3 种算法相比, CPSO 算法的寻优结果最接近最优值. SPSO 算法和 CPSO 算法的收敛轨迹如图 1 所示(测试条件: 维数为 30, 粒子数为 30). 由图 1 可以看出, CPSO 算法的收敛性能优于 SPSO 算法.

表 2　Schwefel 函数的平均寻优值

维数	粒子数	APSO	SPSO	SGA	CPSO
10	15	-1 942.0	-2 466.2	-3 381.2	-3 900.8
	30	-2 772.2	-2 635.6	-3 763.3	-3 971.8
30	15	-6 168.3	-5 811.9	-7 901.5	-11 095.2
	30	-6 996.7	-6 703.6	-9 412.1	-11 925.2

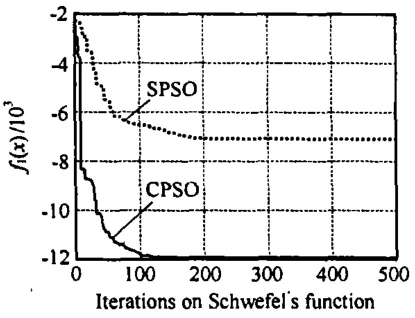

图 1　CPSO 与 SPSO 对 Schwefel 函数测试的寻优曲线

2) Rosenbrock 函数

$$f_2(x) = \sum_{i=1}^{D} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]. \quad (9)$$

Rosenbrock 函数是很难极小化的病态二次函数. 其极小点所在的山谷易于找到, 但要收敛到全局极小点则十分困难.

在测试计算中, 演化代数、个体数目和维数的取值方法同 Schwefel 函数.

表 3　Rosenbrock 函数的平均寻优值

维数	粒子数	APSO	SPSO	SGA	CPSO
10	15	6.359	17.097	1.383e+5	5.849
	30	4.576	5.596	1.198e+5	3.938
30	15	118.114	124.907	5.898e+6	81.033
	30	50.113	50.830	4.290e+6	41.774

表 3 的计算结果表明, 对于单峰 Rosenbrock 函数, 在相同的演化代数条件下, 与粒子群类的算法相比, 遗传算法的寻优能力和收敛结果最差. 粒子群类算法中, CPSO 算法的优化性能最强, APSO 次之. 由优化的结果可以发现, 相同的维数和演化代数, 粒子数为 30 的寻优结果优于粒子数为 15 的结果. CPSO

算法与SPSO 算法收敛轨迹的对比如图2所示(测试条件: 维数为 30, 粒子数为 30). 由图 2 可以看出, CPSO 算法不仅在最终的寻优结果上, 而且在收敛速度上相比于原有的 SPSO 算法都有显著的改善.

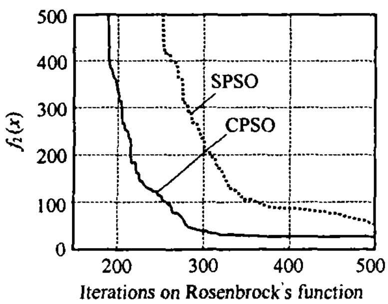

图 2 CPSO 与 SPSO 对 Rosenbrock 函数测试的寻优曲线

3) Schaffer 函数

$$f_3(x) = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5 \quad (10)$$

Schaffer 函数是具有强烈振荡的多峰函数, 一般算法难以得到最优解.

对于 Schaffer 函数, 搜索算法寻到的最优解小于 -0.9999 时, 认为寻优成功. 因此, 通过考察 100 次随机实验中搜索成功的次数来反映算法的优劣.

为便于比较, 演化代数 G_{max} 分别取 200 和 500 两种情况. 从表 4 的优化结果可以看出, CPSO 算法由于引入了混沌序列, 对惰性粒子重新初始化, 搜索成功的次数远远高于其他优化方法. 当演化代数超过某一合适值时, 搜索成功率达到 100%. 该算法与其他优化算法相比, 最大程度地避免了陷入局部最优点, 具有高效的搜索性能.

表 4 Schaffer 函数的成功搜索次数

代数	APSO	SPSO	SGA	CPSO
200	11	17	0	97
500	15	30	2	100

5 结 语

本文针对粒子群优化算法的局部收敛问题, 提出了一种嵌入混沌序列的混合粒子群算法. 测试结果表明, 改进的混合算法解决了粒子群算法的局部收敛问题, 而且通过微粒自适应更新机制确保了全局搜索性能和局部搜索性能的动态平衡. 改进的算法保持了 PSO 算法运算简洁的特点, 在收敛性能上有显著提高.

参考文献(References)

[1] Kennedy J, Eberhart R C. Particle Swarm Optimization [A]. *IEEE Int Conf on Neural Networks*[C]. Perth Australia, 1995: 1942-1948.

[2] 李宁, 刘飞, 孙德宝. 基于带变异算子粒子群优化算法的约束布局优化研究[J]. *计算机学报*, 2004, 27(7): 897-903.
(Li N, Liu F, Sun D B. A Study on the Particle Swarm Optimization with Mutation Operator Constrained Layout Optimization[J]. *Chinese J of Computers*, 2004, 27(7): 897-903.)

[3] 高鹰, 谢胜利. 免疫粒子群优化算法[J]. *计算机工程与应用*, 2004, 6: 4-6.
(Gao Y, Xie S L. Particle Swarm Optimization Algorithms with Immunity[J]. *Computer Engineering and Applications*, 2004, 6: 4-6.)

[4] Kennedy J. Small Worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance[A]. *Proc of the Congress on Evolutionary Computation*[C]. Washington DC, 1999: 1931-1938.

[5] 杨维, 李岐强. 粒子群优化算法综述[J]. *中国工程科学*, 2004, 6(5): 87-93.
(Yang W, Li Q Q. Survey on Particle Swarm Optimization Algorithm[J]. *Engineering Sciences*, 2004, 6(5): 87-93.)

[6] 周驰, 张海兵, 等. 粒子群优化算法[J]. *计算机应用研究*, 2003, 12: 7-11.
(Zhou C, Zhang H B, et al. Particle Swarm Optimization (PSO) Algorithm[J]. *Application Research of Computers*, 2003, 12: 7-11.)

[7] Gwo-Ching Liao, Ta-Peng Tsao. Application Embedded Chaos Search Immune Genetic Algorithm for Short-term Unit Commitment[J]. *Electric Power Systems Research*, 2004, 71(2): 135-144.

[8] 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(1): 416-420.
(Lv Z S, Hou Z R. Particle Swarm Optimization with Adaptive Mutation[J]. *Acta Electronica Sinica*, 2004, 32(1): 416-420.)