

文章编号: 1007-6735(2012)04-0314-04

基于模糊粒子群算法的非线性函数优化

柳 寅, 马 良, 黄 钰

(上海理工大学 管理学院 200093)

摘要: 针对非线性函数优化问题, 提出一种新型的模糊粒子群算法. 该算法基于模糊控制器中输入输出的模糊化处理和粒子群寻优的特点. 算法在 Matlab 2008 环境下编程实现, 针对几个典型复杂的非线性函数进行优化测试. 实现结果表明: 模糊粒子群算法是一种简单有效的算法, 具有良好的有效性和鲁棒性.

关键词: 非线性函数优化; 模糊规则; 粒子群算法; 模糊粒子群算法

中图分类号: N 94; TP 301.6 **文献标志码:** A

Nonlinear Function Optimization Based on Fuzzy Particle Swarm Algorithm

LIU Yin, MA Liang, HUANG Yu

(Business School, University of Shanghai for Science and Technology 200093)

Abstract: For the nonlinear function optimization problem, a fuzzy particle swarm algorithm was proposed based on the fuzzy rules and the characteristics of particle swarm algorithm. The proposed algorithm was programmed in Matlab 2008, and verified through a series of typical nonlinear function optimization problems. The computational results show that the proposed algorithm is a simple and useful algorithm with effectiveness and robustness.

Key words: nonlinear function optimization; fuzzy rules; particle swarm algorithm; fuzzy particle swarm algorithm

粒子群算法 (particle swarm algorithm, PSA) 是一种新型智能优化群体算法^[1-2], 这种算法起源于人们对鸟类觅食行为的研究. 同遗传算法类似 PSA 也是一种基于迭代的优化算法. 目前 PSA 已在函数优化、神经网络优化^[3]、系统识别^[4]等领域有了较广

泛的应用. 但传统 PSA 经常在解决实际问题^[5-8]时, 尤其在解决大规模的问题时容易出现算法过早停滞的缺点, 其导致算法陷入局部最优解. 本文针对传统 PSA 的上述缺点提出了改进的算法: 使用模糊规则^[9-11]引进新的扰动因子改进粒子群算法, 简称

收稿日期: 2012-02-21

基金项目: 国家自然科学基金资助项目 (70871081); 上海市重点学科建设资助项目 (S30504); 上海市研究生创新基金资助项目 (JWCXSL1201)

作者简介: 柳 寅 (1986-), 男, 博士研究生. 研究方向: 智能优化. E-mail: kestral@sina.com
马 良 (联系人), 男, 教授. 研究方向: 智能优化. E-mail: maliang@usst.edu.cn.

模糊粒子群算法(fuzzy particle swarm algorithm, FPSA),并通过在典型函数上的测试表明该算法有比较好的全局优化能力.

1 模糊粒子群算法

1.1 粒子群算法基本原理

在传统PSA中,每个优化问题的解都好比是搜索空间中的一只“鸟”,称其为“粒子”.而被优化的函数决定各粒子的适应值,每个粒子同样还有一个决定它们飞翔的方向和距离的速度因素,这决定粒子追随当前的最优粒子在解空间中搜索.

传统PSA的标准进化方程为

$$\begin{aligned} v(t+1) &= \omega v(t) + c_1 \text{rand}(pbest(t) - x(t)) + c_2 \text{rand}(gbest(t) - x(t)) \\ x(t+1) &= x(t) + v(t+1) \end{aligned}$$

式中, v 为粒子的速度; ω 为惯性权重; rand 为 $[0, 1]$ 之间的随机数; c_1, c_2 为学习因子; $x(t)$ 为第 t 次迭代时粒子的方向.

在每次的迭代过程中,各粒子都通过两个“极值”来更新自己:其一是粒子自身当前迭代过程的最优位置,记为 $pbest$;其二是群体当前迭代过程的最优位置,记为 $gbest$.其中,第 i 个粒子表示为 n 维的向量 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$,即第 i 个粒子的位置为 x_i ,每个粒子代表一个可能的解.

1.2 模糊粒子群算法的基本思路

因为传统PSA在全局极值 $gbest$ 或个体极值 $pbest$ 得到局部最优解时,粒子群不会在解空间中再次进行搜索,而且其它粒子将迅速向局部最优解靠拢,所以使算法容易出现过早收敛导致不能得到最优

解.又因为传统PSA的全局搜索模式相对单一(PSA中仅仅使用全局极值点的信息,而没有加入其它的参考信息,所以使得粒子产生的方式比较单一),这样不利于种群多样性且阻碍算法扩大搜索的范围.

针对上述问题,利用模糊控制器中所使用的模糊规则对传统粒子群算加入了新的扰动因子.在粒子群算法迭代的早期(即迭代计数器 NC 较小的时候),因为此时算法正处于大面积寻优的初步阶段,所以这个时候不应该让每次迭代更新后的 v 过大;而伴随着迭代次数的增加,后面逐步加大 v .到了粒子群算法迭代的后期(即 $NC > 3/4 NC_{\max}$),大幅度增加 v ,其 v 的显著改变对粒子的搜索方向产生较大的影响,这使算法更容易跳出局部最优解,从而更容易获得最优的解.

此外,模糊粒子群算法还利用每次各个粒子所求得的质量价值 $Value$,结合 NC 的大小综合得出干扰因子的大小,而不是每次都仅仅根据 NC 的大小来决定干扰因子.

对于各个粒子所获得的解的 $Value$,综合 NC ,以这两个值作为模糊控制器的模糊输入 $Output$.对这两个量进行模糊化划分和模糊量化,然后,确定生成扰动因子的模糊控制规则,最后对输出的模糊量进行反模糊化,最终作用于每个粒子此次的速度更新量.

1.3 模糊粒子群算法流程

首先确定隶属度函数的形状,这里为了方便计算,两个模糊输入变量的隶属度函数都取等腰三角形,输出变量的隶属度函数为棒形单数值函数,模糊语言数目设定5个,即将模糊输入输出空间平均分割成5个模糊集: S (小)、 $M-$ (较小)、 M (中)、 $M+$ (较大)、 B (大),具体见图1.

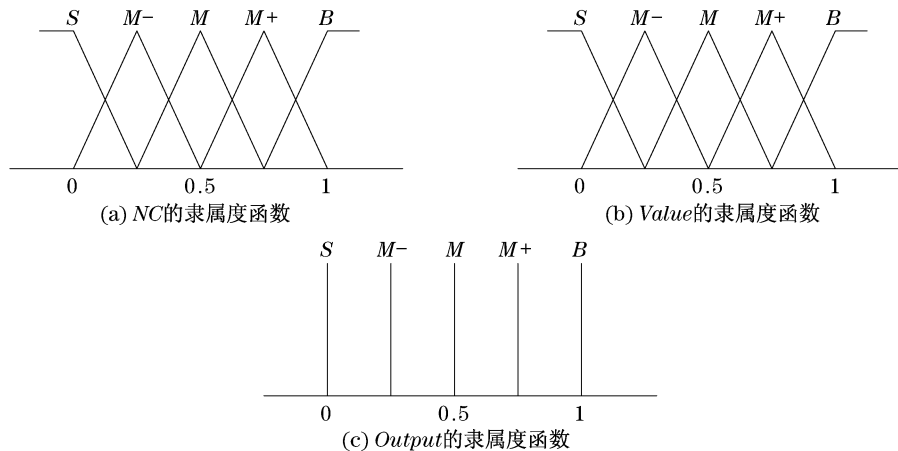


图1 模糊输入输出隶属度函数

Fig.1 Member functions of fuzzy input and output

系统模糊规则采用的形式为
if x is A_i and y is B_i then z is C_i
经过大量的实验比较后,具体决定扰动因子生成的模糊规则如表 1 所示.

表 1 模糊控制规则表
Tab.1 Fuzzy rules table

NC	Value				
	S	M-	M	M+	B
S	Z	Z	S	M-	M
M-	Z	S	M-	M	M+
M	S	M-	M	M+	B
M+	M-	M	M+	B	B
B	M	M+	B	B	B

系统有两个输入,有 5 个模糊值,因此最大的规则基中有 $5 \times 5 = 25$ 条规则.由于后件也有 5 个模糊值,则前件对后件的规则空间为 25×5 矩阵.

推理和模糊化方法采用 Mamdani 推理,重心法.由于输出的隶属度函数是棒形函数,所以重心法在这里就变成了简单的加权平均,由此进一步简化了算法的运算量.

根据模糊规则表中的模糊规则定义,输入的模糊化就是将实际的输入变量从基本论域 $([Input1_{min}, Input1_{max}], [Input2_{min}, Input2_{max}])$, 转换到模糊输入论域 $(0,1)$.

同理,输出的去模糊化就是将模糊输出论域 $(0,1)$ 转换到实际的输出基本论域 $([Output_{min}, Output_{max}])$,模糊控制规则表图形如图 2 所示.

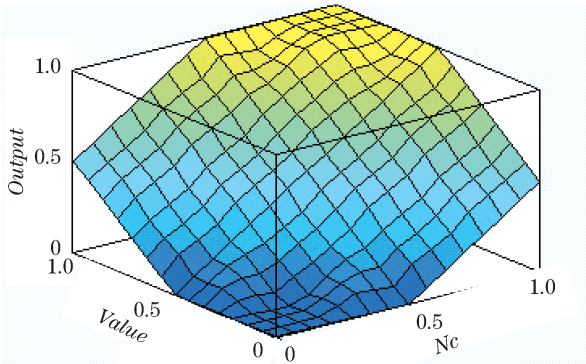


图 2 模糊控制规则表图形
Fig.2 Image of fuzzy rules table

对于具体的输入 (a,b)
 $a \in [Input1_{min}, Input1_{max}]$
 $b \in [Input2_{min}, Input2_{max}]$

经下式转换为 (f_a,f_b) ,其中 $f_a,f_b \in (0,1)$.

$$f_a = \frac{a - Input1_{min}}{Input1_{max} - Input1_{min}} \tag{1}$$
$$f_b = \frac{b - Input2_{min}}{Input2_{max} - Input2_{min}} \tag{2}$$

对于输出具体的模糊输出 $f_c,f_c \in (0,1)$,可转换为

$$c = f_c(Output_{max} - Out_{min}) + Output_{min} \tag{3}$$

其中, $c \in [Output_{min}, Output_{max}]$.

- 模糊粒子群算法流程:
- 步骤 1 随机初始化粒子群中粒子的位置与速度;
 - 步骤 2 将粒子的 $pbest$ 设置为当前位置, $gbest$ 设置为初始群体中最佳粒子的位置;
 - 步骤 3 判断算法停止准则是否满足,如果满足,转向步骤 7,否则执行步骤 4;
 - 步骤 4 按照模糊粒子群算法的更新机制更新各粒子的速度和移动方向,更新方程为
$$v(t+1) = \lambda(\omega v(t) + c_1 rand(pbest(t) - x(t) + c_2 rand(gbest(t) - x(t)))$$
$$x(t+1) = x(t) + v(t+1)$$
其中, λ 为反模糊化后的模糊输出值(扰动因子).
 - 步骤 5 更新所有粒子经历过的最好位置,即全局极值 $gbest$ (如果可行粒子的目标函数值优于 $gbest$ 的目标函数值, $gbest$ 设置为新位置);
 - 步骤 6 判断算法停止准则是否都满足,如果满足则转步骤 7,否则执行步骤 4;
 - 步骤 7 输出 $gbest$,算法停止.

2 算例测试

为了验证算法的可行性和有效性,选取了一系列经典函数进行测试实验.参数设置:种群规模设定为 20~80 个、 $c_1 = c_2 = 2$ 、 $\omega = 0.8$ 、最大迭代次数为 100.实验所用硬件为 AMD 2.0 GHz,2 GB RAM,软件为 Windows XP 和 Matlab 2008.

算例 1 (Griewank 函数)

$$\min Z = \frac{1}{4000} \sum_{i=1}^2 x_i^2 - \prod_{i=1}^2 \left| \frac{x_i}{\sqrt{l}} \right| + 1$$
$$-100 \leq x_i \leq 100$$

此函数为漏斗形状,最小极值点位于图像最中央.采用遗传算法遗传 200 代后的结果为 0.093 22. LINGO 软件所得到的目标值为 0.623 42,Matlab 优化工具箱得到的结果为 0.060 31,模糊粒子群算法

得到的最优解为 $0, (x, y) = (0, 0)$. Griewank 函数图像由图 3 所示.

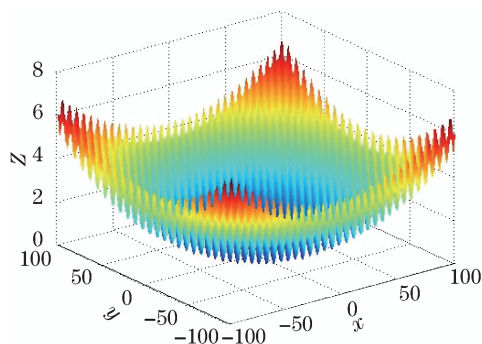


图3 Griewank 函数图像

Fig.3 Image of function Griewank

算例 2 (Schaffer 函数)

$$\max Z = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)]^2}$$

$$x, y \in [-4, 4]$$

混沌优化方法所需计算次数为 1 092, 混沌遗传算法所需计算次数为 458. LINGO 软件所得到的目标值为 0.646 848 8^[11], Matlab 优化工具箱得到的结果为 0.990 3, 模糊粒子群算法得到的最优解为 1, $(x, y) = (0, 0)$. Schaffer 函数图像由图 4 所示.

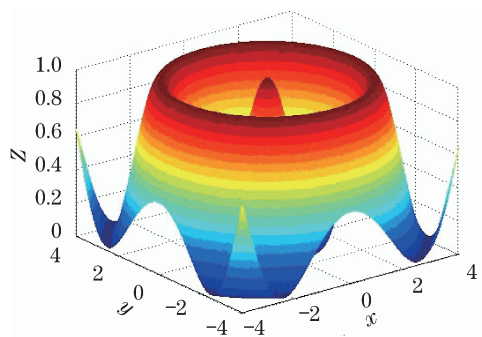


图4 Schaffer 函数图像

Fig.4 Image of function Schaffer

算例 3 (Hansen 函数)

$$\max Z = \sum_{i=1}^5 i \cos((i-1)x + i) \cdot \sum_{j=1}^5 j \cos((j+1)y + j), x, y \in [-10, 10]$$

此函数局部极值点有 760 个. LINGO 软件得到的结果为 12.354 7. Matlab 内部函数所得结果与初值有关, 初值取得好, 则可得到最优解, 有不确定因素的存在^[11]. 遗传算法 100 代内可得到最优解.

模糊粒子群算法可以得到全局最优值为 176.541 793, 并经多次反复运行, 可找到全部最优解 $(-7.589\ 893, -7.708\ 314), (-7.589\ 893, -1.425\ 128), (-7.589\ 893, 4.858\ 057), (-1.306\ 708, -7.708\ 314), (-1.306\ 708, -1.425\ 128), (-1.306\ 708, 4.858\ 057), (4.976\ 478, -7.708\ 314), (4.976\ 478, -1.425\ 128), (4.976\ 478, 4.858\ 057)$. Hansen 函数图像由图 5 所示.

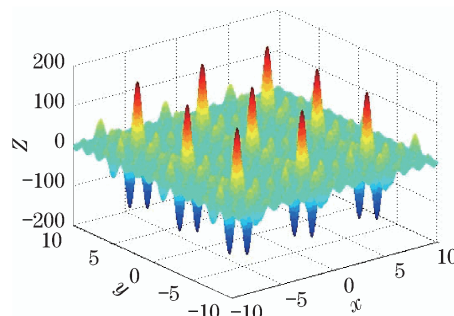


图5 Hansen 函数图像

Fig.5 Image of function Hansen

算例 4 (大海捞针问题)

$$\max Z = \left[\frac{3}{0.05 + (x^2 + y^2)} \right]^2 + (x^2 + y^2)^2, x, y \in [-5.12, 5.12]$$

该问题的全局最优解被最差解包围, 4 个局部极值点为: $(-5.12, 5.12), (-5.12, -5.12), (5.12, -5.12), (5.12, 5.12)$, 函数值为 2748.78. LINGO 和 Matlab 内部函数, 都会落入上述 4 个局部极值点中. 采用有一定技巧的遗传算法在遗传 300 代后才能以 90% 的几率获得最优解, 运行模糊粒子群算法可得到最优解: 3 600, $(x, y) = (0, 0)$. 算例 4 函数图像由图 6 所示.

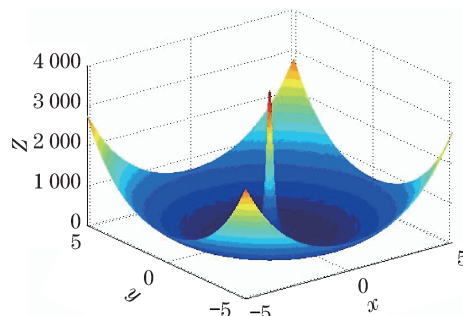


图6 算例 4 函数图像

Fig.6 Image of function 4

从以上的算例可以看出, 模糊粒子群算法具有更有效地获取全局最优解的能力.

(下转第 322 页)