

基于 MATLAB 的粒子群优化算法程序设计

吴建生, 秦发金

(柳州师范高等专科学校 数学与计算机科学系, 广西 柳州 545004)

摘 要: 阐述了粒子群算法的基本原理, 探讨了在 MATLAB 环境中实现粒子群算法的编程方法, 构建粒子群算法工具箱函数, 通过仿真示例验证了该方法的有效性, 表明它能够对函数进行全局优化。

关键词: 粒子群; 优化算法; MATLAB 程序设计

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1003-7020(2005)04-0097-04

1 引言

群体智能算法 (Swarm Intelligence Algorithm, SIA) 的研究开始于 20 世纪 90 年代, 其基本思想是模拟自然界生物的群体行为来构造随机优化算法^[1,2,3], 通常单个自然界的生物并不是智能的, 但是整个生物群体却表现出处理复杂问题的能力, 群体智能算法就是模仿这些生物的团体行为并把它应用在人工智能问题中, 其中粒子群优化算法 (Particle Swarm Optimization, PSO) 就是群体智能算法的一种, 它是由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 在 1995 年提出的, 其基本思想是对鸟群、鱼群的觅食过程中的迁徙和聚集的行为模拟, 并利用了生物学家 Frank Heppner 的生物群体模型^[4,5,6]。PSO 算法是一类基于群体智能的随机优化技术, 相对遗传算法而言, 二者都是基于群体的迭代搜索, 但是 PSO 算法没有交叉、变异算子, 粒子群优化算法是通过个体之间的协作来搜寻最优解, 它利用了生物群体中信息共享的思想, 其概念简单、易于实现, 同时又有深刻的智能背景, 既适合科学研究, 又特别适合工程应用。因此, PSO 一提出, 就引起了众多学者的关注, 并在短短几年的时间里出现了大量的研究成果^[7,8]。

2 PSO 基本原理

PSO 算法是基于群体智能理论的优化算法, 群体中的粒子在每次迭代搜索的过程中, 通过跟踪群体 2 个极值: 粒子本身所找到的最优解 P_{best} 和群体找到的最优解 G_{best} 来动态调整自己位置和速度^[5,6], 完成对问题寻优, 对于如下的函数优化问题

$$\begin{cases} \min f(x_1, x_2, \dots, x_n) \\ s.t. R_{1j} \leq x_j \leq R_{2j}, j = 1, 2, \dots, n \end{cases} \quad (1)$$

其计算步骤如下^[11]:

(1) 对粒子群中粒子的位置和速度进行随机初始化。设定群体的规模为 N , 则随机生成如下矩阵

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} & v_{11} & v_{12} & \dots & v_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} & v_{21} & v_{22} & \dots & v_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & \dots & x_{Nn} & v_{N1} & v_{N2} & \dots & v_{Nn} \end{bmatrix}, \quad (2)$$

其中 $\{x_{ij}, i = 1, 2, \dots, N, j = 1, 2, \dots, n\}$ 表示群体中 i 粒子的位置为 j , v_{ij} 是对应它的速度, 二者均为区间 $[R_{1j}, R_{2j}]$ 上均匀分布的随机数。

(2) 计算每个粒子的适应度 (目标函数值)。

(3) 计算粒子所经历的最好位置 $pbest_i(t) = (x_{i1}, x_{i2}, \dots, x_{in})$, 也就是粒子所经历过的具有最好适应度的位置, 由下式确定

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & f(x_1(t+1), x_2(t+1), \dots, x_n(t+1)) < pbest_i(t), \\ x_i(t+1) & f(x_1(t+1), x_2(t+1), \dots, x_n(t+1)) \geq pbest_i(t). \end{cases} \quad (3)$$

计算群体中所有粒子经历过最好位置, 即全局最好位置

$$Gbest_i(t) = \max \{f(Pbest_1(t)), f(Pbest_2(t)), \dots, f(Pbest_N(t))\}, \quad (4)$$

(4) 依据下式对粒子的速度和位置进行进化

$$v_{ij}(t+1) = v_{ij} + c_1 r_1 (Pbest_i(t) - x_{ij}(t)) + c_2 r_2 (Gbest_i(t) - x_{ij}(t)), \quad (5)$$

收稿日期: 2005-09-02

作者简介: 吴建生 (1974—), 陕西咸阳人, 硕士, 讲师, 研究方向: 神经网络应用及智能优化算法; 秦发金 (1967—), 男, 广西临桂人, 副教授。

基金项目: 广西教育科学“十五”规划课题

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (6)$$

其中 $c_{1,2}$ 为加速度常数 (学习速率), $r_{1,2}$ 为 $[0, 1]$ 均匀分布的随机数。

(5) 判断结束条件, 目标函数的适应度达到足够好或者进化到预先设定的代数, 否则返回步骤 (2), 继续进行。

3 用 MATLAB 实现粒子群算法

MATLAB 是 MathWorks 公司的产品, 是一个功能强大的科学与工程计算软件, 集成了计算、可视化和程序编制等功能, 其使用直观、简洁并符合人们思维习惯的代码给用户提供了一个友好的开发环境^[9-10], 利用 MATLAB 矩阵运算的强大功能编写粒子群算法程序有很好的优势。

3.1 粒子群的初始化

initpop 函数的功能是实现群体的初始化, popsize 表示群体的大小, dimsize 表示粒子维数, 它由变量的维数决定, xmin 是变量的下限, xmax 是变量的上限。

```
function pop = initpop(popsiz, dimsize)
pop = unifrnd(xmin, xmax, popsize * dimsize);
```

3.2 计算粒子的适应度和确定群体的 Pbest 和 Gbest

calobjvalue 函数的功能是计算目标函数的适应度, 其公式是采用本文的示例仿真, 用户可以根据不同的优化问题予以修改。

```
function objvalue = calobjvalue(pop)
for i = 1: popsize
    obfuct1 = sin(sqrt(pop(i, 1)^2 + pop(i, 2)^2)) - 0.5;
    obfuct2 = (1.0 + 0.001 * (sqrt(x(i, 1)^2 + x(i, 2)^2))^2);
    objvalue(i, 1) = 0.5 + obfuct1/obfuct2;
end
Pbest = pop(:, 1: dimsize);
[ggBest, xindex] = max(objvalue);
xtemp = pop(xindex, 1: dimsize);
Gbest = xtemp;
```

3.3 粒子速度和位置的更新

粒子速度和位置的更新是基于式 (5) 与 (6), 并生成新的粒子群体。

```
function pop2 = renew(pop)
for t = 1: popsize
    for dimIndex = 1: dimsize
        w = wmax - (wmax - wmin) * (generation / maxgeneration);
        sub1 = Pbest(t, dimIndex) - pop(t, dimIndex);
        sub2 = Gbest(1, dimIndex) - pop(t, dimIndex);
        tempV = w * pop(t, dimsize + dimIndex) + c1 * unifrnd(0, 1) * sub1 + c2 * unifrnd(0, 1) * sub2;
        if tempV > speedmax
            pop(t, dimsize + dimIndex) = speedmax;
        elseif tempV < (-speedmax)
            pop(t, dimsize + dimIndex) = -speedmax;
        else
            pop(t, dimsize + dimIndex) = tempV;
        end
        tempposition = pop(t, dimIndex) + pop(t, dimsize + dimIndex);
        if tempposition > xmax
            pop(t, dimIndex) = xmax;
        elseif tempposition < xmin
            pop(t, dimIndex) = xmin;
        else
            pop(t, dimIndex) = tempposition;
        end
    end
end
```

3.4 粒子 Pbest 和 Gbest 的更新

粒子在进化过程中依据其适应度, 调节个体最好位置 Pbest和群体最好位置 Gbest

```
function [ Pbest Gbest] = regulate( pop)
for i= 1: popsize
    obfuct1= sin( sqrt( x( i 1) ^2+ x( i 2) ^2) ) ^2- 0.5
    obfuct2= ( 1.0+ 0.001* ( sqrt( x( i 1) ^2+ x( i 2) ^2) ) ^2);
    objvalue( i 1) = 0.5+ obfuct1/obfuct2
    obfuct3= sin( sqrt( pBest( i 1) ^2+ pBest( i 2) ^2) ) ^2- 0.5
    obfuct4= ( 1.0+ 0.001* ( sqrt( pBest( i 1) ^2+ pBest( i 2) ^2) ) ^2);
    pvalue( i 1) = 0.5+ obfuct3/obfuct4
end

obfuct1= sin( sqrt( gBest( 1) ^2+ gBest( 2) ^2) ) ^2- 0.5
obfuct2= ( 1.0+ 0.001* ( sqrt( gBest( 1) ^2+ gBest( 2) ^2) ) ^2);
objvaluetemp= 0.5+ obfuct1/obfuct2

for i= 1: popsize
    if objvalue( i 1) < pvalue( i 1)
        PBest( i 1: dimsize) = pop( i 1: dimsize);
    end
    if objvalue( i 1) < objvaluetemp
        GBest= pop( i 1: dimsize);
        xtemp= pop( i 1: dimsize);
    end
end
end
```

4 示例仿真

利用上述思路在 MATLAB中开发 PSO 工具箱, 以 J.D. Schaffer提出的函数^[11]

$$\begin{cases} \min f(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{X_1^2 + X_2^2} - 0.5}{[1.0 + 0.001(X_1^2 + X_2^2)]^{2.1}} \\ s.t. -100 \leq x_i \leq 100 \quad (i = 1, 2) \end{cases} \quad (7)$$

为示例仿真, 按照上述方法编写程序, 它的最优解 $\min(f(X^*)) = f(0, 0) = 0$ 此函数在距全局最优点大约 3.14范围内存在无穷多个局部极小将其包围, 并且该函数强烈振, 一般的算法难以得到最优解, 利用 PSO 工具箱计算它, 其运行界面如下:

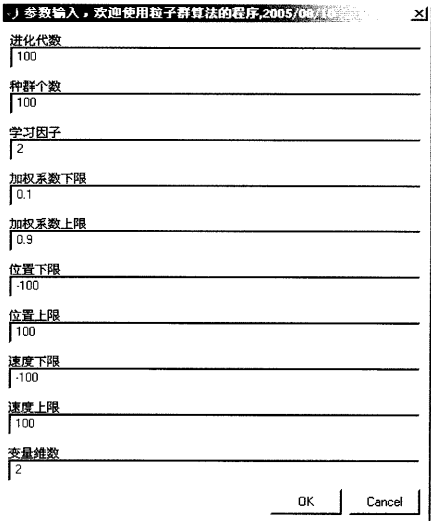


图 1 基于 MATLAB的 PSO 算法运行界面

在运行过程中, 为了评价算法的性能, 采用 DeJong提出的在线性能^[12] (on-line performance)和离线性能 (off-line performance)来衡量算法性能的优劣, 在环境 e 下策略 s 的在线性能 $X_e(s)$ 定义为:

$$X_e(s) = \frac{1}{T} \sum_{t=1}^T f_e(t), \quad (8)$$

式中 $f_e(t)$ 是在环境 e 下第 t 时刻的平均目标函数值或平均适应度。算法的在线性能指标表示了算法从开始运行一直到现在为止的时间段内算法性能值的平均值,它反映了算法的动态性能。在环境 e 下策略 s 的离线性能 $X_e^*(s)$ 定义为:

$$X_e^*(s) = \frac{1}{T} \sum_{t=1}^T f_e^s(t), \tag{9}$$

式中 $f_e^s(t)$ 是在环境 e 下,在 $[0, 1]$ 时段内最好的目标函数值或最大适应度。算法的离线性能表示了算法运行过程中各进化代的最佳性能的累积平均,它反映了算法的收敛性能。

在示例仿真中我们取种群个数为 100 进化代数 100 学习因子 (加速度常数) 2 加权系数上限为 0.9 和下限为 0.1 位置下限为 -100 上限 100 速度设置与位置相同。图 2 为本文算法求解 J.D. Schaffer 函数过程中算法的在线性能和离线性能。算法运行 10 次,其全部收敛到最优解 $\min(f(X^*)) = f(0, 0) = 0$ 图 3 是 PSO 工具箱计算示例仿真函数在进化过程中它的适应度变化情况。

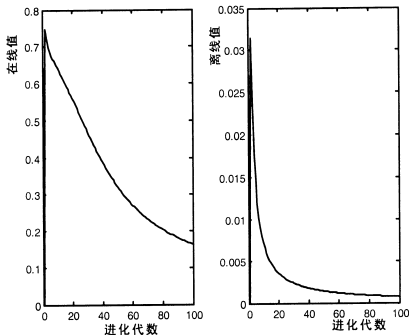


图 2 粒子群进化过程在线和离线性能

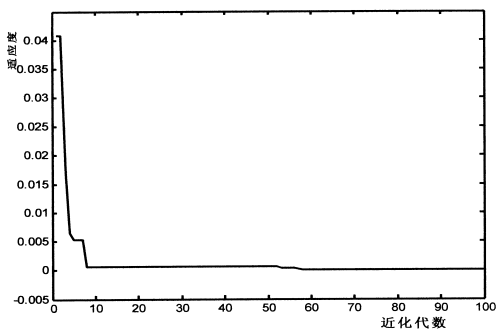


图 3 粒子进化过程适应度的变化

5 结语

我们利用 MATLAB 强大的计算、绘图以及界面设计功能,开发了基于 MATLAB 的粒子群优化算法工具箱,设计界面友好、操作方便,具有很强的通用性,利用它可以方便地解决一些粒子群算法应用问题,可以很直观地看出各种参数的确定对进化结果有何影响,故对学习和应用粒子群算法可以起到事半功倍的作用。

参考文献:

[1]曾建潮,介婧,崔志华.微粒群算法[M].科学出版社,2004.
[2]E. Bonabeau, M. Dorigo, G. Theraulaz. Inspiration for optimization from social insect behavior[J]. Nature 2000, 406(6): 39—442
[3]E. Bonabeau, M. Dorigo, G. Theraulaz. Swarm intelligence from natural to artificial systems[M]. New York: Oxford Univ. Press 1999
[4]J.Kennedy, R. C. Eberhart Particle swarm optimization[A]. Proceeding IEEE International Conference on Neural Networks[M]. 1995: 1942—1948.
[5]张荣沂. 一种新的集群优化方法——粒子群优化算法[J]. 黑龙江工程学院学报(自然科学版), 2004, 18(4): 34—37
[6]杨维,李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004 6(5): 87—92
[7]R. C. E. Schart Y. Shi Evolving Artificial Neural Networks[A]. Proceeding of International Conference on Neural Networks and Brain[M]. Piscataway NJ IEEE Press 1998 5—13.
[8]李宁,邹彤,孙德宝. 带时间窗车辆路径问题的粒子群算法[J]. 系统工程理论与实践, 2004, 24(4): 131—135.
[9]张志涌. 精通 MATLAB6.5[M]. 北京: 航空航天大学出版社, 2004
[10]阮沈勇,王永利,桑群芳. MATLAB 程序设计[M]. 北京: 电子工业出版社, 2004.
[11]王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.
[12]周明,孙树栋. 遗传算法原理及其应用[M]. 国防工业出版社, 2004.

(责任编辑:梁文杰)

A Design of Particle Swarm Optimization with MATAB

WU Jian-sheng QIN Fa-jing

(Department of Mathematics and Computer Science, Liuzhou Teachers College, Liuzhou Guangxi 545004, China)

Abstract The principle of particle swarm optimization has been presented and its realization with MATAB has been discussed, and the particle swarm optimization tool library function was made. A function optimization problem has been given to demonstrate the global optimization function-ability of the MATLAB program.

Key words particle swarm ptimization; MATLAB; programming