

MATLAB环境下遗传算法优化工具箱的应用^{*}

刘万林, 张新燕, 晁勤

(新疆大学 电气工程学院, 新疆 乌鲁木齐 830008)

摘 要: 用 MATLAB 语言及 MATLAB 语言编制的优化工具箱进行优化设计具有语言简单、函数丰富、用法比较灵活、编程效率高等特点。本文对遗传算法和基于 MATLAB 的遗传算法优化工具箱 (GAOT) 作了简要的介绍, 分析了优化工具函数, 并结合非线性、多峰值函数问题的优化实例, 说明了遗传算法是一种具有良好的全局寻优性能的优化方法。

关键词: MATLAB; 遗传算法; 优化; 全局寻优

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1000-2839(2005)03-0357-04

The Application of Genetic Algorithm Optimization Toolbox in MATLAB

LIU Wan-lin, ZHANG Xin-yan, CHAO Qin

(College of Electrical Engineering, Xinjiang University, Urumqi, Xinjiang 830008, China)

Abstract The optimization design in MATLAB and Matlab optimization toolbox have simple language, abundant functions flexible method and high programming efficiency. The paper introduces genetic algorithm (GA) and genetic algorithm optimization toolbox and analyses the optimization toolbox function. The function optimization problem of nonlinear and multi peak has been given to demonstrate that genetic algorithm is a better global optimization method.

Key words MATLAB; Genetic Algorithm; Optimization; Global optimization

Matlab 语言是一种高效率的用于科学工程计算的高级语言, 与 Basic Fortran Visual C 等语言比较, Matlab 语言的语法规则简单、更加贴近人的思维方式, 易学易用。Matlab 语言有着丰富的由世界著名专家、学者开发出的各种工具箱, Matlab 的优化工具箱提供了对各种优化问题的一个完整的解决方案。遗传算法优化工具箱就是其中之一, 遗传算法 (GA) 是一类借鉴生物界自然选择和遗传机制的随机优化搜索算法, 其主要特点是群体搜索策略和群体中个体之间的信息交换。搜索不依赖于梯度信息。由于不受函数约束条件 (如连续性、可微性、单极值) 的限制, 因而具有广泛的适应能力。它尤其适用于处理传统搜索方法难以解决的复杂和非线性问题, 因此, 采用 MATLAB 语言设计的遗传算法优化工具箱将它应用于实际中, 不仅具有简单、易用、易于修改的特点, 而且为解决许多传统的优化方法难以解决的象非线性、多峰值之类的复杂问题提供了有效的途径, 为遗传算法的研究和应用提供了很好的应用前景。

1 遗传算法

1.1 遗传算法 (Geneticalgorithms GA) 是由美国 Michigan 大学的 JohnHolland 教授在 20 世纪 60 年代提出的, 它是一种自然适应优化方法, 该算法是基于自然遗传和自然优选机理形成的一种全局寻优方法。遗传算法是将问题的求解表示成“染色体”, 将它们置于问题的“环境”中, 根据适者生存的原则, 从中选择出适应环境的“染色体”进行复制, 即再生 (reproduction, selection), 通过交叉 (crossover)、变异 (mutation) 两种基因操作产生出新一代更适合环境的“染色体”群, 这样一代代不断改进, 最后收敛到一个最适合环境的个体上 (当然也有其他的收敛准则), 求得问题的最佳解。由于最好的染色体不一定出现

* 收稿日期: 2004-09-17

作者简介: 刘万林 (1978-), 男, 硕士研究生, 从事电力系统综合自动化方面的研究。

在最后一代,开始时保留最好的染色体,如果在新的种群又发现更好的染色体,则用它代替原来的染色体,进化完成后,这个染色体可以看作是最优化的结果.遗传算法几乎渗透到从工程到社会科学的诸多领域,必须要编制遗传算法的程序进行计算,作为使用者希望找一个现成的程序,而 MATLAB的遗传算法工具箱正好满足要求.

1.2 遗传算法的基本构成要素和 GA 的流程

输入参数: 染色体个数 N ,交叉概率 P_c ,变异概率 P_m ;通过初始化过程产生 N 个染色体;计算所有染色体的评价函数;根据评价函数抽样选择染色体;对染色体进行交叉和变异操作;重复若干次(下一代的代数)计算评价函数、选择、交叉和变异.

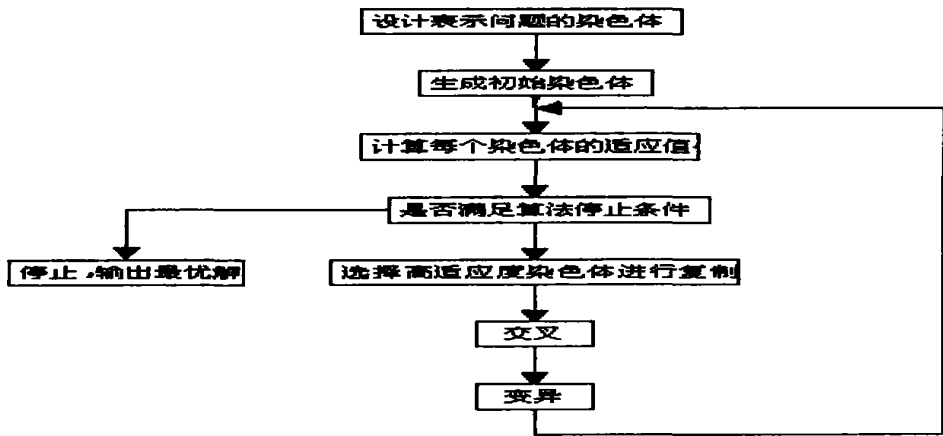


图 1 GA 的流程

1.3 GA 有如下 3 个基本算子: ① 再生 (reproduction/selection).再生算子从群体中按某一概率选择个体,某个体 X_i 被选择的概率 P_i 与其适应值成正比.最通常的实现方法是轮盘赌 (roulettewheel)模型.② 交叉 (crossover).交叉算子将被选中的 2 个个体的基因链按概率 P_c 进行交叉,生成 2 个新的个体,交叉位置是随机的.其中 P_c 是一个系统参数,即交叉概率.③ 变异 (mutation).变异算子按一定概率 P_m 将新个体的基因链的各位进行变异,对二值基因链 (0, 1 编码)来说即是取反. P_m 也是一个系统参数,即变异概率.以上各种算子的实现方法是多种多样的,而且许多高级算子正不断提出,以改进 GA 的某些性能.由于 GA 的性能具有一定的脆弱性 (brittleness),因此 GA 本身的参数 (即系统参数)的选取对 GA 的运行效果有很大影响.

1.4 系统参数的选取一般遵循以下原则:① 种群数目 N .种群数目会影响 GA 的有效性. N 太小,GA 会很差或根本找不出问题的解,因为太小的种群数目不能提供足够的采样点; N 太大,会增加计算量,使收敛时间延长.一般种群数目在 20~ 160 之间比较合适.② 交叉概率 P_c .此参数控制着交叉操作的频率. P_c 太大,会使高适应值的结构很快破坏掉; P_c 太小,搜索会停滞不前.一般 P_c 取 0.25~ 0.75.③ 变异概率 P_m .它是增大种群多样性的第二因素. P_m 太小,不会产生新的基因块; P_m 太大,会使 GA 变成随机搜索.一般 P_m 取 0.01~ 0.20.

2 遗传算法工具箱 (GA Toolbox)

2.1 遗传算法函数

遗传算法工具箱 GAOT 包括了许多实用的函数,这些函数按照功能可以分为以下几类: 主界面函数、选择函数、演化函数、其它的一些终止函数、二进制表示函数、演示程序等等.

2.2 遗传算法函数的参数

MATLAB 的遗传算法工具箱核心函数 GAO TV 5 其主程序 ga. m 提供了遗传算法工具箱与外部的接口,在 MATLAB 环境下,执行 ga 并设定相应的参数,就可以完成优化,它的格式如下:

(1) function [pop]= initializega(num, bounds, eevalFN, eevalOps, options)—初始种群的生成函数; 输出参数]pop—生成的初始种群; [输入参数]num—种群中的个体数目; bounds—代表变量的上下界的矩阵; eevalFN—适应度函数; eevalOps—传递给适应度函数的参数; options—选择编码形式 (浮点编码或是二进制编码) [precision F- or- B],如 precision—变量进行二进制编码时指定的精度 F- or- B—为 1时选择浮点编码,否则为二进制编码,由 Precision指定精度.

(2)function [x, endPop, bPop, traceInfo] = ga(bounds, evalFN, evalOps, startPop, opts, … termFN, termOps, selectFN, selectOps, xOverFNs, xOverOps, mutFNs, mutOps).
[输出参数]x endPop bPop traceInfo; [输入参数]bounds evalFN evalOps startPop opts termFN termOps selectFN selectOps xOverFNs xOverOps mutFNs mutOps.

3 应用实例

[问题 1]求 $f(x)= 1/[(x- 0.2)^2+ 0.01]+ 1/[(x- 0.8)^2+ 0.04]- 4$ 的最大值,其中 $-1 \leq x \leq 2$.

[分析 1]选择二进制编码,种群中的个体数目为 10,二进制编码长度为 20,交叉概率为 0.6,变异概率为 0.1.

(1)编写目标函数存储为 fbgalw. m文件.
function[sol, val]= fbgalw(sol, options)
x= sol(1);
val= 1/[(x- 0.2)^2+ 0.01]+ 1/[(x- 0.8)^2+ 0.04]- 4;

(2)调用主函数 ga. m程序如下:
clc
fplot(1/[(x- 0.2)^2+ 0.01]+ 1/[(x- 0.8)^2+ 0.04]- 4; [- 1 2])
hold on
initPop= initializega(10, [- 1 2], fbgalw);
plot (initPop(: , 1), initPop(: , 2), 'b* ')
xlabel('x '); ylabel('f(x) ');
hold on
[x endPop bpop trace]= ga([- 1 2], fbgalw ; [], initPop, [1e- 6 1 1], 'max GenTerm'; 80, 'normGeomSelect'; [0 1], ['arithXover '], [2], 'nonUnifMutation'; [2 80 3]);
plot (endPop(: , 1), endPop(: , 2), 'r* ')
figure(2)
plot(trace(: , 1), trace(: , 3), 'y- ')
hold on
plot(trace(: , 1), trace(: , 2), 'r- ')
xlabel('generation '); ylabel('Fitness ');
legend('解的变化 ' ; '种群平均值的变化 ');

运行结果如图 2和图 3所示,图 2中标有“*”记号的点为初始值,标有“+ ”的点为最优值.最优解为 $X^*= 0.200\ 38, f_{max}= 98.501\ 409$.

[问题 2]求 $f(x)= x+ 9^* \sin(5x)+ 7^* \cos(4x)$ 的最大值,其中 $0 \leq x \leq 9$

[分析 2]选择二进制编码,种群中的个体数目为 10,二进制编码长度为 20,交叉概率为 0.95,变异概率为 0.08.

(1)编写目标函数存储为 fitness. m文件.

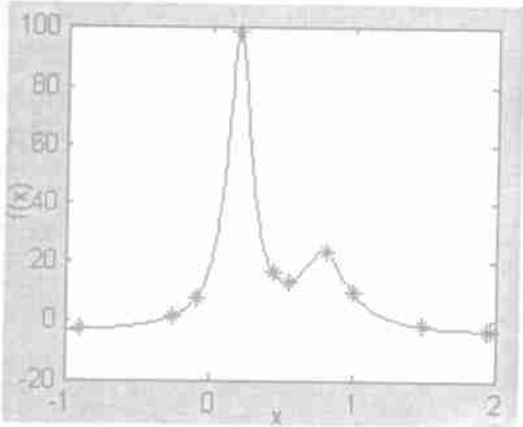


图 2 100次迭代后的寻优结果

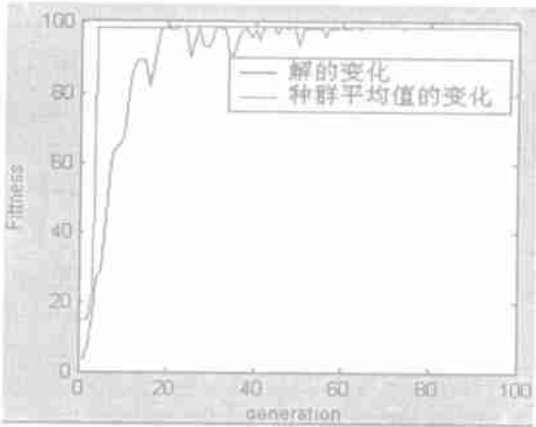


图 3 遗传算法的寻优性能

```
function[sol,eval]= fitness(sol,options)
x= sol(1);
eval= x+ 9* sin(5* x)+ 7* cos(4* 5x);
(2)调用主函数 ga.m程序
```

运行结果如图 4和图 5所示,图 4中标有“*”记号的点为初始值,标有“+”的点为最优值.最优解为 $X^* = 7.8572$, $f_{\max} = 23.855$.

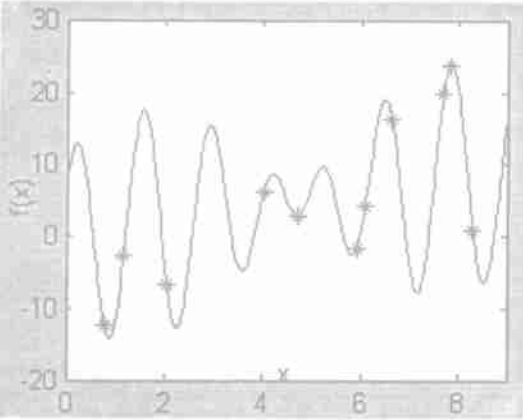


图 4 25次迭代后的寻优结果

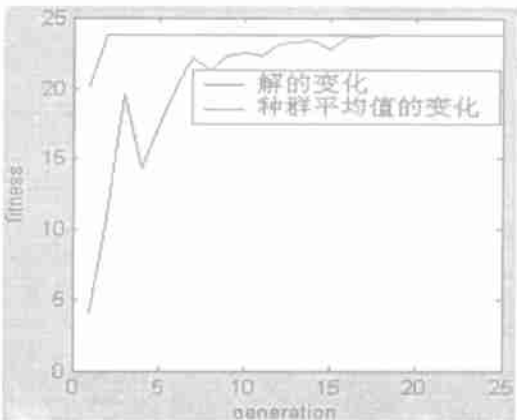


图 5 遗传算法的寻优性能

由上两例从寻优结果和寻优性能图可以看出遗传算法对非线性、多峰值函数完全可以避免陷入局部最优而找到全局最优解,从解的变化可以看出迭代次数不须太多就能达到最优解.

4 结 语

随着遗传算法作为一种全局优化算法在各个领域的应用越来越广泛,笔者在 MATLAB环境下应用遗传算法及采用 MATLAB语言编制的遗传算法优化工具箱对传统优化算法难以实现全局最优解的非线性、多峰值函数最值问题进行了优化,结果表明对于非线性、多峰值函数的寻优问题,遗传算法不仅不会陷入局部最优点,有较强的搜索能力、收敛速度快、精度也较高,而且其用法也比较灵活,完全避免了一般优化算法的局部最优、收敛较慢等问题的不足,并取得了令人满意的效果.

参考文献:

[1] C. R. Houck, J.JonesandM. Kay. Agenetical gorithm for function optimization: A Matlab implementation[M]. ACM Transactionson Mathematical Software, 1996.
[2] 张志涌,刘瑞桢,杨祖英.精通和掌握 MATLAB[M].北京:北京航空航天大学出版社,1998.
[3] 飞思科技产品研发中心. MATLAB6. 5辅助优化计算与设计 [M].北京: 电子工业出版社,2003.

责任编辑: 陈 勇