

# 基于Matlab 的遗传算法的工具箱的研究及改进

姚文俊

(中南民族大学 电子与信息学院 湖北 武汉 430074)

**摘要:** 介绍了使用Matlab 语言编程实现的遗传算法, 并建立遗传算法工具箱。通过仿真实验说明该工具箱具有运行稳定、结构灵活、扩展方便等特点, 能直观显示改进后的GA, 比标准GA 具有收敛速度快、不易陷入局部极小、精度高等特性。

**关键词:** 遗传算法; Matlab; 遗传算法工具箱; 仿真

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1004-373X (2004) 09-036-03

## Study and Improvement of Genetic Algorithm Toolbox Based on Matlab

YAO Wenjun

(College of Electronics and Information Engineering, South-central University for Nationalities, Wuhan, 430074, China)

**Abstract** In this paper, the genetic algorithm adopting theMatlab is achieves and further, genetic algorithm for optimization toolbox is set up. By the simulation experiment, the toolbox can provide a stable, flexible and extendible platform for applying and developing genetic algorithm. The improved GA can directly display the faster convergence, higher precision and better solution-finding rate than the standard GA in the toolbox.

**Keywords** genetic algorithm; Matlab; genetic algorithm for optimization toolbox; simulation

遗传算法 (Genetic Algorithm, GA) 是美国Michigan大学Holland 教授于1965 年首先提出来的, 他是一种基于达尔文进化论的优胜劣汰、适者生存和群体进化学说的全局搜索算法<sup>[1]</sup>。而Matlab 作为一种拥有高性能数值计算能力的通用科技计算机语言在其简单易用的操作环境中集成了数值分析、矩阵计算、图视能力、可视化建模仿真和实时控制能力, 适合多学科和多部门的发展<sup>[2]</sup>。根据Matlab 的强大的数值计算能力、大量的内建辅助函数和开放、可扩展的体系结构这些性质, 将Matlab 引入遗传算法, 在Matlab 平台上开发遗传算法工具箱, 可帮助人们更好地认识 and 了解GA, 并为进一步开发GA 提供基础和帮助。

### 1 遗传算法简介

遗传算法借助于计算机编程, 一般是将待求问题表示成串 (或染色体), 即为二进制码或数码串, 从而构成一群串, 并将他们置于问题的求解环境中, 根据适者生存的原则, 从中选择出适应环境的串进行复制 (Reproduction), 且通过交换 (Crossover)、变异 (Mutation) 两种基因操作产生新一代更适应环境的串群, 经这样一代代地不断变化, 最后收敛到一个最适应环境的串上, 而求得问题的最优解。其步骤主要分成以下6 步:

#### (1) 选择编码方式和编码

GA 在执行求解之前, 首先要选择合适的编码方式,

将待求解问题的所有参量编码成一个定长的字符串, 设串长为 $L$ 。目前, 常使用的编码方式主要是二进制码。

#### (2) 产生初始群体

即由计算机随机产生 $N$  个初始字符串, 每个字符串称为一个个体, 这 $N$  个个体就构成了一个群体。GA 正是以这 $N$  个字符串作为初始点开始迭代计算。

#### (3) 计算适应度函数值

适应度函数反映了个体对环境适应能力的强弱, 根据对他的计算值, 可以很好地控制个体生存的机会, 以体现适者生存的自然法则。一般来说, 对于不同的问题, 适应度函数的定义方式也不尽相同。

#### (4) 复制

一个群体由 $N$  个个体组成, 这其中哪个个体保留用以繁殖后代, 哪个被淘汰, 是根据对他们各自的适应度函数值的计算来决定的。个体的适应度函数值大, 说明其对环境的适应能力强, 他也就有更多的机会被保留下来用以复制后代; 反之, 则其被淘汰。可见, 复制依据的原则是适应度大的个体为下一代贡献一个或多个后代的概率大。

#### (5) 交换

对于依据复制的原则而被选中, 用于繁殖的2 个个体, 随机选择位置 $i(1 \leq i \leq l)$ , 交换2 个字符串位置 $i$  左边 (或右边) 的部分, 产生2 个新个体, 这2 个新个体分别组合了其父辈个体的特性。显然, 交换新个体的目的在于产生新的基因组合, 形成新的个体, 而不是重复上一代的同

一个群体。

#### (6) 变异

首先在群体中随机地选择一个个体, 对于选中的个体以一定的概率随机地改变字符串中某位上的字符的值。当采用二进制码串时就是将选中位上的1变为0或0变为1。

GA正是经历了上述6个步骤并执行6至3步的循环操作, 才求出问题的最优解。

## 2 基于Matlab的GA优化工具箱

遗传算法的优化工具箱<sup>[3]</sup> (Genetic Algorithm for Optimization Toolbox, GAOT) 主要要考虑到上述6个基本操作及终止条件、二进制和十进制的相互转换等一些其他的操作。下面介绍具体的实现步骤。

### 2.1 GA主函数

基于Matlab的GA的基本调用函数如下:

```
[x, endPop, bPop, traceInfo] = ga(bounds, evalFN, evalParams, params, startPop, temFN, temParams, selectParams, xOverFNs, xoverFNs, xOverParams, mutFNs, mutParams)
```

其中主函数的等号前面是输出参数, 后面是输入参数, 即GA的基本操作, 分别是一些m文件。GAOT就是通过编写这些m文件来实现和改进GA。下面是输入参数的介绍。

### 2.2 初始化和终止函数

initializega函数实现群体的初始化, popsize表示群体的大小, bounds表示染色体的长度, 取决策变量采用的编码长度; termination函数实现进化代数的评价, 函数采用最大进化代数和最优进化代数两种方式。函数形式如下:

```
function[pop] = initializega(populationsize, bounds, evalFN, evalOps, options)
function[done] = maxGenTem(op, bPop, endPop);
function[done] = optmaxGenTem(op, bPop, endPop)
```

### 2.3 评价函数

评价函数是GA运行的推动力, 通过函数调用可以在算法搜索中求函数的个体适应值和最大值。一个例子如下:

```
function[x, val] = gaDemoEval(x, parameters)
val = x(1) + 10 * sin(5 * x(1) + 7 * cos(4 * x(2)));
```

调用以下函数运行的个体适应值的最大值:

```
BestX = ga([0 10; 0 - 10], gaDemoEval)
```

### 2.4 遗传操作

#### (1) 选择

选择是决定哪些个体可以进入下一代。常用的轮盘赌选择法, 这种方法根据适应值比例来选择个体, 较易实现。函数为:

```
function[newpop] = roulette(oldPop, options)
```

#### (2) 交叉

交叉过程选取2个个体作为父代, 产生2个新的子代。函数为:

```
function[c1, c2] = crossover(p1, p2, bounds, ops)
```

#### (3) 变异

变异由一个父代产生一个子代。函数为:

```
function[child] = mutation(parent, bounds, ops)
```

## 2.5 二进制和浮点数的转换

二进制转换浮点数函数是:

```
function[fval] = b2f(bval, bounds, bits)
```

浮点数转换二进制函数是:

```
function[bval] = f2b(fval, bounds, bits)
```

## 3 仿真实例

有了基于Matlab的GA工具箱GAOT, 进行函数的优化和进行GA的改进就很容易了, 只需改写函数m文件格式。以下是一些仿真例子:

(1) 求函数 $f(x) = x + 10\sin 5x + 7\cos 4x$ 在区间 $[0, 9]$ 的极大值。

按照以上的工具箱函数, 取 $\text{popsize} = 10$ ,  $\text{maxGenTem} = 25$ , 几何规划排序选择, 算术交叉, 非统一变异, 其中 $p_c = 0.95$ ,  $p_m = 0.08$ 。仿真实验结果如图1所示,  $x$ 轴表示遗传算法进化的代数,  $y$ 轴表示种群中染色体的适应度值。图1中实线代表种群个体中当前最优值的轨迹, 虚线则代表种群中个体的平均值的轨迹。

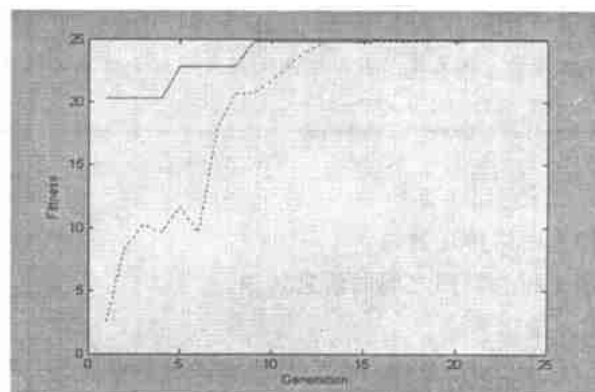


图1 函数的测试结果

#### (2) 改进后的遗传算法的仿真

将遗传算法进行改进, 改进的方法编成m文件, 然后在工具箱中应用。以下是Rosenbrock函数:

$$f(x) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$$

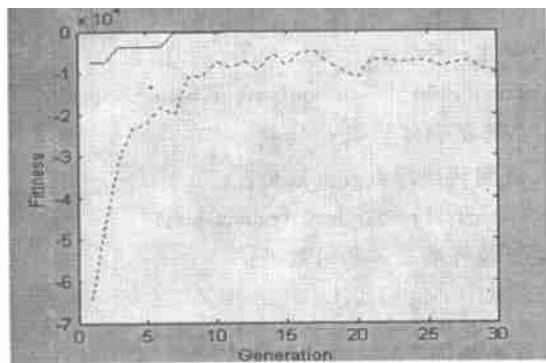
他们全局最小值是位于一个非常狭窄的山谷中的点 $[1, 1, \dots, 1]^N$ , 即便是在 $N = 2$ 的情况下, Rosenbrock函数的解空间有非常多的狭窄的通道, 导致很难获得全局最优值。当取更大值的时候, 情况会更加复杂。这里提出了3条改进的方案:

对初始种群进行正交化设计来选取。

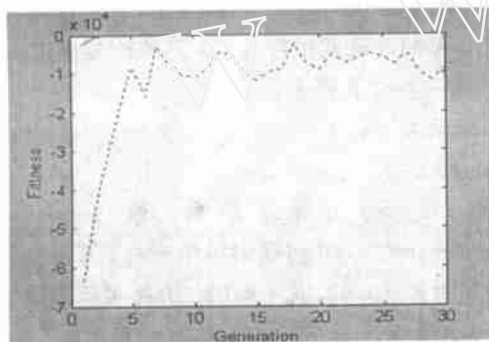
变固定的交叉率和变异率为自适应变化。

将局部搜索能力强的爬山法与 GA 接合, 引进自适应的正交局部搜索算子。

标准 GA 和改进 GA 仿真实验结果比较如图 2 所示。



(a) 标准遗传算法



(b) 基于自适应正交局部搜索算子的遗传算法  
图 2 Rosenbrock 函数的测试结果比较图

#### 4 结 语

基于 Matlab 的遗传算法工具箱提供了一个标准的、可扩展的、简单的算法, 其利用 Matlab 的强大的矩阵运算能力, 使使用者可以避免维护遗传算法种群和染色体的数据结构的重编程工作, 将精力集中在遗传算法的改进和具体问题的应用中。

本文在仿真实验中既利用了 GAOT 本身提供的 m 函数文件, 也将改进的 GA 算法编成 m 文件在 GAOT 中运行, 通过仿真图能够直观地反映 GA 与其他传统寻优方法相比, 具有明显优势, 同时直观显示改进 GA 比标准 GA 具有收敛速度快、不易陷入局部极小、精度高等特性。

#### 参 考 文 献

- [1] 周明, 孙树栋. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 1999.
- [2] 张志涌, 徐彦琴. Matlab 教程 [M]. 北京: 北京航空航天大学出版社, 2001. 4.
- [3] Christopher R Houck, Jeffery A Joines, Michael G Kay. A genetic algorithm for function optimization: a matlab implementation.

作者简介 姚文俊 男, 中南民族大学电子信息工程学院讲师。

(上接第 35 页)

#### (3) 网络传送数据

用于相关部门之间的信息传送。

#### 3.2.2 输出方式

(1) 报表输出各部门在处理完相关业务后要进行整理, 做出相关报表, 如图 4 所示。

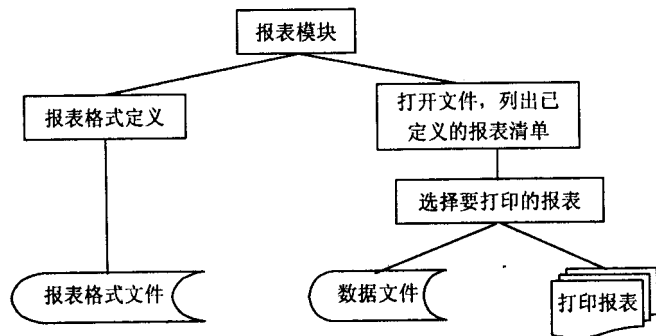


图 4 报表生成设计

#### (2) 图形输出

对于决策系统最重要产生有利于决策者做出决策的相

关图形, 例如产品销售曲线图, 员工业务销售情况分析图以及各部门的业绩对比图等。

#### 4 结 语

本系统在 Windows NT 环境下基本实施。在系统设计方面, 采用生命周期法和快速原型法相结合, 在已有产品的基础上以快速原型法为主。在设计思想方面, 采用面向对象设计, 将着眼点放在用户身上。在组织实施方面, 采用专业化和产品化的实施体制, 确保产品能够安全维护和持续升级。

#### 参 考 文 献

- [1] 薛化成. 管理信息系统 (第 3 版) [M]. 北京: 清华大学出版社, 1999.
- [2] 刘瑞新, 汪远征. Visual Basic 程序设计教程 [M]. 北京: 机械工业出版社, 2000.