

# 遗传算法在 MATLAB 环境中的实现

汪秉文 范 犇 康小海  
(华中理工大学)

**摘要** 探讨了在 MATLAB 环境中实现遗传算法仿真的方法, 并以一个简单的求函数最值的问题作为遗传算法的应用实例, 说明遗传算法的全局寻优性及用 MATLAB 实现仿真的可行性。

**关键词** 遗传算法; MATLAB; 全局寻优

中图法分类号 TP 301.6

遗传算法 GA (Genetic Algorithm) 是近几年发展起来的一种崭新的全局优化算法。它借用了生物遗传学的观点, 通过自然选择、遗传和变异等作用机制, 使每个个体的适应性提高<sup>[1]</sup>。由美国 Mathwork 公司于 1967 年推出的 Matrix Laboratory (缩写为 MATLAB) 软件包, 是一种功能强、效率高便于进行科学和工程计算的交互式软件包。在此环境下, 所解问题的 MATLAB 语言表述形式和其数学表达形式相同, 不需要按传统方法编程<sup>[2]</sup>。在 MATLAB 环境下编制一个简单的遗传算法工具库 (SGA), 就可以利用 MATLAB 强大的仿真功能, 进行遗传算法的各种仿真实验。

## 1 一个基本的遗传算法

遗传算法是将问题的求解表示成“染色体”, 从而构成一群“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<sup>[3]</sup>。图 1 给出了 GA 的流程图<sup>[4]</sup>。

GA 有如下 3 个基本算子:

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 (roulette wheel) 模型。

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。

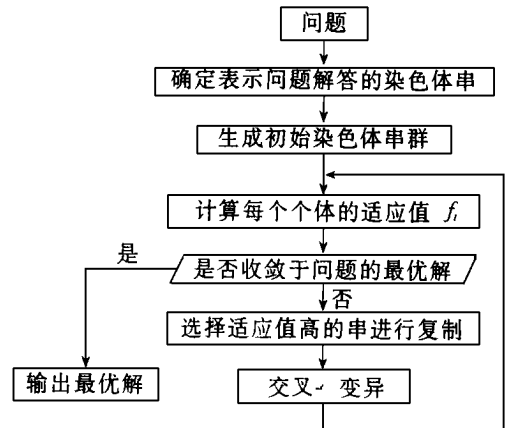


图1 GA 的流程图

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。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 SGA 库

前已述及, MATLAB 中最重要的成分是函数,下面简要地介绍一下 SGA 库中主要的函数及变量。在 SGA 库中,经常要使用的变量是  $P_{op}$ ,它代表一个种群,是各种遗传算子操作的对象。 $P_{op}$  本身是一个维数为  $\text{popsize} \times \text{itemsize}$  的矩阵。之所以这样做是考虑到 MATLAB 处理矩阵的强大能力。矩阵的每一行是一个维数为  $\text{itemsize}$  的向量(数组),分别代表一个染色体。由于向量维数  $\text{itemsize}$  理论上可以无限增大,这就保证了染色体的长度可以根据需要无限增长。种群的大小是  $\text{popsize}$ ,即染色体的个数。

SGA 中包含如下 3 个实现基本遗传算法的函数:

Crossop,即交叉算子,以概率  $P_c$  在两染色体上的随机位置交换子串,格式为  $P_{op} = \text{crossop}(P_{op}, P_c)$ ,其中参数  $P_c$  代表交叉概率。

Muta,即变异算子,以概率  $P_m$  对染色体上任一基因进行干扰,格式为  $P_{op} = \text{muta}(P_{op}, P_m)$ ,其中  $P_m$  代表变异概率。

Repro,即繁殖(再生/选择)算子,基本的选择策略是采用轮盘赌模型。轮盘经任意旋转停止后指针所指向区域被选中,所以  $f_i$  值大的个体被选中的概率就大。该函数格式为  $[P_{op}, f_p] = \text{repro}(P_{op}, f_p)$ 。其中  $f_p$  代表适值(fitness)向量,维数等于种群大小  $\text{popsize}$ ,  $f_p$  的第  $i$  个元素对应于  $P_{op}$  的第  $i$  行向量,即种群的第  $i$  个染色体的适值。

要实现遗传算法的功能,还需要以下几个重要的函数:

Initpop,即初始化种群。函数格式为  $P_{op} = \text{initpop}(\text{popsize}, \text{itemsize}, \text{'m'})$ 。式中  $\text{popsize}$  及  $\text{itemsize}$  的含义如前所述,字符  $\text{'m'}$  是  $\text{method}$  的缩写,其含义是产生初始种群的染色体串的方式。在本例中,此参数取  $\text{'b'}$  代表染色体取二进制串;此参数取  $\text{'d'}$  代表染色体取十进制串。

Decode,即解码函数。对应于不同的问题域,需定义不同的解码方式(编码过程由手工完成)。

Fof,即适值填充函数。函数格式为  $f_p = f_{of}(P_{op})$ ,其功能是根据种群计算种群的适值向量,这也是一个和问题求解域紧密联系的函数,需要用户根据具体情况编制相应的程序。

以下函数使用户在调试遗传算法时感到更加方便:

**Best**, 即取最佳个体函数。函数格式为  $[sam, fit] = best(P_{op}, f_p)$ , 其功能是找到种群中适值最大的个体(individual)作为样板 sam, 并输出其适值 fit。

**Worst**, 即取最差个体函数。函数格式为  $[sam, fit] = worst(P_{op}, f_p)$ , 其功能是找到种群中适值最小的个体(individual)作为样板 sam, 并输出其适值 fit。

**Average**, 即求平均适值函数。函数格式为  $aver = average(f_p)$ , 其功能是求取种群的平均适值。

**Sortpop**, 即种群排序函数。函数格式为  $[P_{op}, f_p] = sortpop(P_{op}, f_p)$ 。其功能是将种群中的个体按其对应适值的大小从小到大进行重新排列。

**Seed**, 即播种函数。其形式为  $P_{op} = seed(P_{op}, p)$ , 其功能是在原种群中以比例  $p$  随机输入新的种群, 以替代原种群中适值最低的相应个体。 $p$  为  $0 \sim 1$  之间的小数。当种群的个体开始收敛而 GA 仍未得到较好结果时, 采用该函数可以使 GA 进入新的搜索空间, 并最终寻找到最佳值。

### 3 应用实例

笔者以一个简单的求函数最值问题为例, 说明 GA 的功能及 SGA 的可用性。函数形式取双峰函数  $f(x) = 1/[(x - 0.3)^2 + 0.01] + 1/[(x - 0.9)^2 + 0.04] - 6$ , 自变量范围取为  $[-1, 2]$ 。由图 2 知, 当  $x = 0.3$  时, 函数  $y = f(x)$  取最大值 96.5; 当  $x = 0.9$  时,  $y = f(x)$  取局部极大值 21.7。

采用 GA 寻优, 每个个体的基因解码为自变量  $x$ , 适值函数取为  $400K/(C_{max} -$

$f(x))$ ,  $K$  为 114.3,  $C_{max} = 100$ 。种群大小  $N$  取 20, 交叉率  $P_c$  取 0.6, 变异率  $P_m$  取 0.1。以下是遗传 200 代后获得的结果。

图 3 显示了种群中最佳个体适应值增长的情况。由于在再生算子中保留了父代种群中适值最高的个体, 所以在图 4 中几乎看不到适应值减小的现象。

图 4 显示了 GA 的搜索过程。在一次具有代表性的实验中, 第一代种群中适应值最高的染色体经解码后其值为 0.55, 正好处在函数图像中波谷的  $O$  点(见图 2), 此时 GA 如向右搜索, 则有可能找到一个局部极大点  $B$ , 但 GA 向左搜索则会成功地找到全局最大点  $A$ , 充分显示了 GA 的全局寻优功能。

### 4 结束语

遗传算法作为一种全局优化算法正在得到广泛的应用, 笔者在 MATLAB 环境下用 MATLAB 语言编制了一个基本的 SGA 库, 并以求函数最大值为应用例子说明了 SGA 库的应用及遗传算法的全局优化效果。由于 MATLAB 软件包强大的仿真功能以及良好的可扩充性, 在 MATLAB 环境下使用 SGA 库, 是对遗传算法进行仿真研究的一个有力工具。

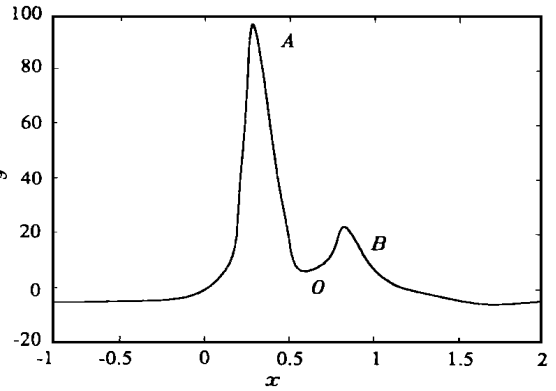


图 2 双峰函数示意图

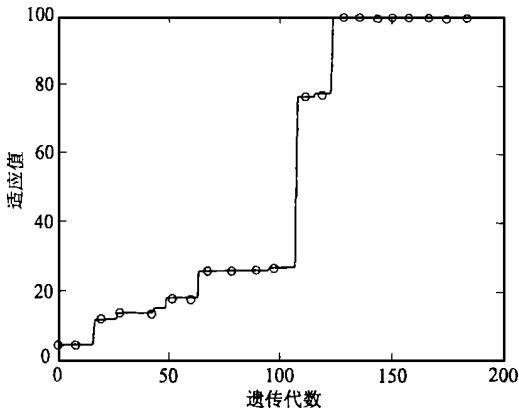


图3 适应值增长情况

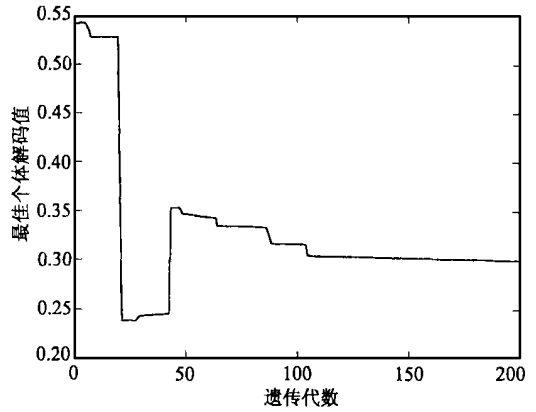


图4 搜索过程图

## 参 考 文 献

- 1 张晓绩, 戴冠中. 一种新的优化算法——遗传算法. 控制理论与应用, 1995, 12(3): 265 ~ 271
- 2 张培强. MATLAB 语言——演算纸式的科学工程计算语言. 合肥: 中国科学技术大学出版社, 1997.
- 3 Goldberg D E. Genetic Algorithm in Search, Optimization and Machine Learning. MA: Addison-Wesley Reading, 1989
- 4 陈根社, 陈新海. 遗传算法的研究与进展. 信息与控制, 1994, 23(4): 215 ~ 222

## Realization of Genetic Algorithm in MATLAB

Wang Bingwen

Fan Zhun

Kang Xiaohai

**Abstract** A method to realize genetic algorithm in MATLAB is discussed. A function optimization problem is presented to demonstrate the feasibility of this method as well as demonstrating the global optimization functionality of genetic algorithm.

**Key words** genetic algorithm ; MATLAB ; global optimization

**Wang Bingwen** Assoc. Prof. ; Dept. of Auto. Contr. Eng., HUST, Wuhan 430074, China.

(责任编辑:李道文)