

MATLAB 遗传算法工具箱的设计*

谢勤岚 陈红

(中南民族大学电子信息工程学院 武汉 430074)

陶秋生

(武汉数字工程研究所 武汉 430074)

摘 要

在简要分析遗传算法要素的基础上,介绍了基于 MATLAB 的遗传算法工具箱的设计。最后给出了一个用设计的 MATLAB 遗传算法工具箱的求函数极值的应用实例。

关键词:遗传算法 MATLAB 函数优化

中图分类号:TP311

Design and Application of Genetic Algorithm Toolbox based on MATLAB

Xie Qinlan Chen Hong

(College of Electronic and Information Engineering, SCUFN, Wuhan 430074)

Tao Qiusheng

(Wuhan Digital Engineering Institute, Wuhan 430074)

Abstract: Based on the simple analysis of genetic algorithm, the designing of a MATLAB-based genetic algorithm toolbox is introduced. An example using the designed toolbox to maximize a function is given at last.

Key words: genetic algorithms, MATLAB, function optimizing

Class number: TP311

1 引言

遗传算法(genetic algorithm, 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(如连接性、可微性、单极性)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,可广泛应用于机器学习、优化设计、自适应控制、规划设计

和人工生命等领域,是 21 世纪有关智能计算中的关键技术之一。其特点确定了它是一种通用的优化算法,因此设计通用的遗传算法程序是可行的^[1,2,3,5]。

MATLAB 可应用于高性能数字计算的工程计算环境,它将数值分析、矩阵计算和绘图集成在一个易于使用的环境中^[4]。用户定义函数是简单的解释结构的文本文件。因此,用 MATLAB 设计遗传算法工具箱具有简单、易用、易于修改的特点。

我们设计的基于 MATLAB 的遗传算法

* 收到本文时间:2003 年 2 月 17 日

工具箱(GA Toolbox based on MATLAB),构造合理,扩展方便,并能与 MATLAB 中的其它工具箱结合使用。为遗传算法的研究和应用提供了有力的工具。

2 遗传算法(GA)的四个关键部分

遗传算法是一个以适应度函数(或目标函数)为依据,通过对群体中的个体施加遗传操作,实现群体内个体结构重组的迭代处理过程。图 1 是基本的遗传算法框图^[1,3]。遗传算法一般由四个关键部分组成:编码机制、适应度函数、遗传算子、运行参数。

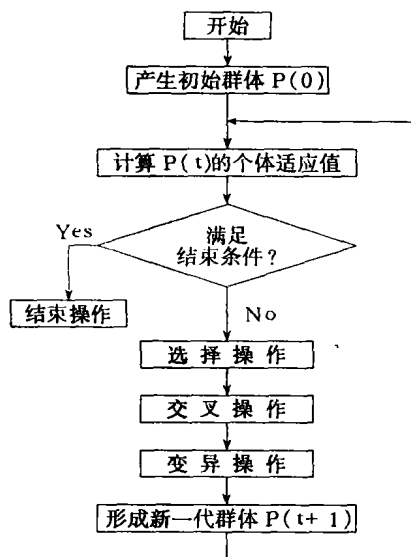


图 1 遗传算法操作流程框图

2.1 编码机制

遗传算法不是对研究对象直接进行讨论,而是通过某种编码机制把对象统一赋予由特定符号(字母)按一定顺序排成的串,这种赋值过程就是编码。编码是影响遗传算法效率的重要因素。串的集合构成总体(群体),个体(染色体)就是串。GA 的码可以具有广泛的含义。在优化问题中,一个串对应于一个可能解;在分类问题中,串可解释为一个规则,即串的前半部为输入或前件,后半部为输出或后件、结论,等等。这也正是 GA 有广泛应用的重要原因。

2.2 适应度函数

遗传算法的一个特点是它仅使用所求问题的目标函数值就可以得到下一步的有关的搜索信息。而对目标函数值的使用是通过评价个体的适应度来体现的。

适应度用来度量群体中各个个体在优化算法中有可能达到或接近于或有助于找到最优解的优良程度。适应度较高的个体遗传到下一代的概率就较大;而适应度较低的个体遗传到下一代的概率就相对小一些。度量个体适应度的函数称为适应度函数(Fitness Function)。对优化问题,适应值函数就是目标函数。

2.3 遗传算子

遗传算子是遗传操作的重要内容,主要的算子有三种:选择、交叉、变异。

遗传操作算子中变异与交叉算子是染色体数据结构的对应。交叉算子根据交叉率(P)将种群中的两个个体随机地交换某些基因,从而产生新的基因组合(个体)。交叉运算是产生新个体的主要方法,它决定了遗传算法的全局搜索能力。对于一维数组类型的染色体,常用的交叉算子包括一点交叉、两点交叉、多点交叉和均匀交叉。对于实数编码的染色体,还有算数交叉和混合交叉等交叉算子。

变异算子模仿了生物繁殖过程中的基因突变。它根据变异率将个体染色体编码串中的某些基因座上的基因值用该基因座的其他等位基因来替换,从而形成一个新的个体。变异运算是产生新个体的辅助方法,它决定了遗传算法的局部搜索能力。对二进制染色体常用的变异算子有翻转变异和交换变异,对实数编码的染色体通常采用高斯变异。

选择算子(或称复制算子)根据种群中各染色体适应度来分配各个个体的繁殖机会,与染色体的结构无关。复制算子反映了进化论中“适者生存”的理论。选择运算的主要目的是为了避免基因缺失,提高全局收敛性和

计算效率。经典的选择复制算子是采用轮赌法复制算子,此外还有排序选择、随机选择、期望值模型选择等复制算子。

2.4 运行参数

遗传算法中需要选择的运行参数包括种群的大小、个体编码串长度、交叉率、变异率、最大进化代数等,这些参数对遗传算法的性能都有重要影响。对于具体问题而言,衡量参数设置恰当与否,要根据多次运行的收敛情况和解的质量来判断。如果调整参数难以有效地提高遗传算法的性能,则往往需要借助对基本遗传算法的改进,改进的手段可以是多方面的,如适应度比例调整、引入自适应交叉率和变异率,尝试其它的遗传操作,也可以采用一些混合方法等。

3 遗传算法工具箱的设计

设计遗传算法工具箱的目的是为研究和应用遗传算法提供性能可靠、使用方便、扩展性强的运算平台。采用 MATLAB 作为编写遗传算法工具箱的平台,能充分利用 MATLAB 强大的矩阵运算能力,并且易与其它工具结合。设计的遗传算法工具箱的结构(局部)如图 2 所示。

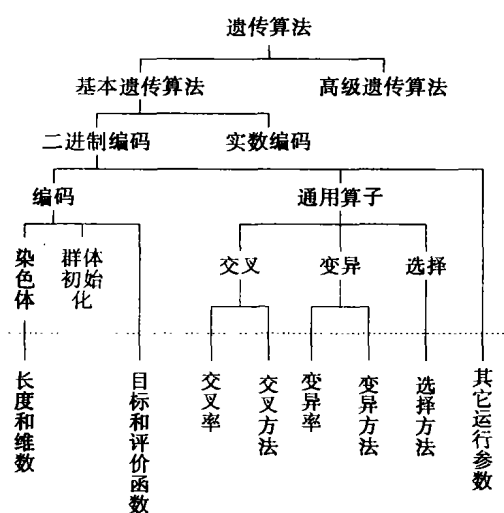


图 2 遗传算法工具箱的结构(局部)

工具箱中的函数有两类:一类是供工具

箱中的其它函数调用的函数;另一类是供使用者调用的函数,这些函数必须有调用格式和参数说明。同时,工具箱还要提供使用说明、要求使用者编写的函数的编写和调用格式。

在遗传算法程序中,染色体的数据结构、对应的交叉和变异算子、复制算子、代沟与具体问题无关,编写的遗传算法工具包提供了对上述要素的支持。这些与图 2 的虚线上部相对应。对与具体问题相关的适应度函数和终止条件,在遗传算法工具包中给出了对应函数的编写和调用格式。遗传算法参数在遗传算法工具包中被集中在参数向量中,可以根据算法设计需要进行修改。这些与图 2 的虚线下部相对应。

4 计算实例

这里选择文献[2]中给出的问题,求下面函数的最大值:

$$f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2)$$

其中 $-3.0 \leq x_1 \leq 12.1$ 及 $4.1 \leq x_2 \leq 5.8$ 。假定对每个变量要求的精度是小数点以后第四位。采用二进制编码的方式,每个染色体由两部分构成,第一部分确定 x_1 ,需要 18 位;第二部分确定 x_2 ,需要 15 位。因此,染色体(解向量)的总长度为 $m = 18 + 15 = 33$ 位。群体规模为 20。交叉概率和变异概率分别确定为 0.25 和 0.01。利用设计的遗传算法工具箱(采用了精英算子),运行 150 代后,得到最优解: $x_1 = 11.628$, $x_2 = 5.2249$, $y = 38.3438$ 。

5 结论

基于 MATLAB 的遗传算法工具包充分利用了 MATLAB 强大的矩阵运算能力,使用者可以避免繁重的编程和维护工作,将精力集中在对遗传算法的改进和具体问题的应用中。遗传算法工具包能够自由地与 MAT-

LAB 中的已有功能和工具包相结合,为使用者将遗传算法与其他学科研究相结合提供了便利的工具。

在本文实现的遗传算法工具箱内,只提供了对优化计算和参数优化设计的支持。现在 MATLAB 已提供了对面向对象方法和高级图形技术的支持,如将它们引入到遗传算法工具箱中,将使工具的应用更广泛,功能更强大,使用更方便。

(上接第 46 页)

高可用性的硬件设计主要是通过增加冗余设备来完成的。硬件的冗余设计可以是整机的备份,也可以是针对单个组件进行备份,如对处理器、内存、电源等的单独备份。通常有三种冗余配置方案:

(1)2N 模式:一个运行态的节点,一个备用的节点。

(2) $N+1$ 模式: N 个运行态的节点,一个备份的节点。

(3) $N+M$ 模式: N 个运行态的节点, M 个备份的节点。

5.2 高可用性的软件设计

为了达到软件设计的高可用性,可以通过下面的方法达到。

(1)软件设计采取主备(active/standby)模式。主备模式有两种,主备软件同时运行和主备软件不是同时运行,而是主运行态软件向备份软件进行经常的状态更新。第二种优越性在主运行态软件发生错误时,备用软件将在最近状态更新的地方开始运行,这样避免了主备软件由于相同的错误而同时坏死。

(2)主备软件运行在不同的节点上。避

- 参考文献**
- [1]周明,孙树栋.遗传算法原理及应用.国防工业出版社,1999
 - [2]Z. 米凯利维茨[美].演化程序——遗传算法和数据编码的结合.科学出版社.2000
 - [3]王小平,曹立明.遗传算法——理论、应用与软件实现,西安交通大学出版社,2002
 - [4]程卫国等. MATLAB5.3 应用指南.人民邮电出版社 1999
 - [5]潘正君,康立山,陈毓屏.演化计算.清华大学出版社,1998

免一个节点垮掉时使得整个主备软件都坏死。

6 结论

本文对 3G 核心网软件体系架构和关键技术进行分析,提出的软件架构层次清晰,功能完善,上层软件与软件之间以及软件与底层的操作系统和硬件几乎处于独立位置,因此,适应核心网的发展趋势,易于软件的移植和底层硬件的升级,便于硬件的更新换代。同时,针对特定的电信级设备,提出了软件的可移植性和系统的高可用性的设计。

参考文献

- [1]通信软件内部文献
- [2]3GPP TS 23.002 v4.3.0(2001-06)
- [3]3GPP TR 21.905: "3G Vocabulary"
- [4]3GPP TS 23.101: "General UMTS Architecture"
- [5]胡捍英 杨峰义,《第三代移动通信系统》,人民邮电出版社,2001.8
- [6]Roger S. Pressman, "Software Engineering, A Practitioner's Approach, Fourth Edition", 机械工业出版社,黄柏素 梅宏译,1999.10
- [7]叶敏,《程控数字交换与现在通信网》,北京邮电大学出版社 1998.2