

Matlab 遗传算法工具箱的设计

郭 强, 赵晓华, 孙 亮

(北京工业大学 电子信息与控制工程学院, 北京 100022)

摘 要: 介绍了使用 Matlab 语言实现遗传算法, 建立遗传算法工具箱 (GAs toolbox). 开发的遗传算法工具箱支持二进制和浮点式编码方式, 并且提供了多种选择、交叉、变异的方法, 用以提高遗传算法的性能. 使用工具箱对一系列非线性、多峰值函数问题进行了测试, 显示了优良的性能. 通过测试, 发现浮点式遗传算法的寻解效率和精确度都优于二进制算法以及传统的寻优方法.

关键词: 遗传算法; 函数优化; Matlab

中图分类号: TP 273.5

文献标识码: A

文章编号: 0254-0037(2001)03-0348-05

遗传算法是演化计算的一个重要分支, 是一种基于自然选择和基因遗传学原理的优化搜索方法. 遗传算法吸收了自然生物系统“适者生存, 优胜劣汰”的进化原理, 从而使它能够提供一个在复杂空间中进行鲁棒搜索的方法, 为解决许多传统的优化方法难以解决的优化问题提供了新的途径. 与传统的搜索和优化过程相比, 遗传算法在计算机上模拟生物的进化过程和基因操作, 因而不需要对象的特定知识, 也不需要对象的搜索空间是连续可微的, 具有全局寻优的能力.

Matlab 作为一种拥有高速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、符号计算、图视能力、文字处理、可视化建模仿真和实时控制能力, 适合多学科、多部门的发展需求. Matlab 本身提供的大量内建辅助函数对于处理函数优化问题十分有用, 而且相当全面、简便易用, Matlab 的数值计算功能很强, 另外, Matlab 开放、可扩展的体系结构允许用户开发自己的应用程序^[1]. Matlab 的这些性质为开发遗传算法工具箱提供了一个统一和熟悉的操作环境. 为了更好地利用遗传算法的强大功能, 作者在 Matlab 下开发了遗传算法工具箱 (GAs toolbox), 为 Matlab 增加了一个有利的工具. 并可帮助人们认识、了解遗传算法, 同时也为进一步开发遗传算法的应用提供了基础和有益的帮助.

1 遗传算法^[2]

遗传算法是由美国 Michigan 大学的 John Holland 教授在 60 年代提出的, 该算法是基于自然遗传和自然优选机理的寻优方法. 自然遗传和自然优选来自于达尔文的进化论学说. 该学说认为, 在生物进化过程中, 任一动植物经过若干代的遗传和变异, 使之能够适应新的环境, 是优胜劣汰的结果; 这种自然遗传思想也适用于求解优化问题. 遗传算法采用选择 (selection)、交叉 (crossover) 和变异 (mutation) 运算来实现这一自然法则的模拟; 其演算流程如图 1 所示.

选择 其过程为, 基于个体对环境的适应度 (由 $f / \sum f$ 决定; 其中 f 是对象的适配值, $\sum f$ 是种群所有对象的适配值之和), 决定哪个个体被复制. 选择意味着由较高适配值的个体, 被复制的概率也比较大.

交叉 指在选择个体成员的过程中, 按一定概率随机选择匹配对; 然后两个匹配的个体位串根据随机选取的交叉点进行交叉繁殖, 产生一对新的个体串. 采用二进制编码, 交叉可以描述如下 ($|$ 是交叉点):

个体 1 11011|00100110110, 后代 1 11011|11000011110;

收稿日期: 2000-07-07.

作者简介: 郭强 (1978-), 男, 本科生 (已毕业).

个体2 11011|11000011110, 后代2 11011|00100110110.

变异 是以一定的概率改变位串的某个位置的值. 采用二进制编码, 交叉可以描述如下:

原来的后代1 1101111000011110, 变异的后代1 1100111000011110;

原来的后代2 1101100100110110, 变异的后代2 1101101100110110;

在遗传算法中先寻找一个基本可行解, 然后对这个基本可行解进行遗传操作, 通过选择、交叉和变异3个算子在串编码空间进行搜索, 根据系统输出的结果性能对编码串进行评价, 利用随机技术进行复制, 同时选择交叉点, 对编码串进行交叉, 得到新的下一代编码串, 在这个过程中允许在变异概率内对编码串进行变异. 这一过程重复执行直到某种条件被满足.

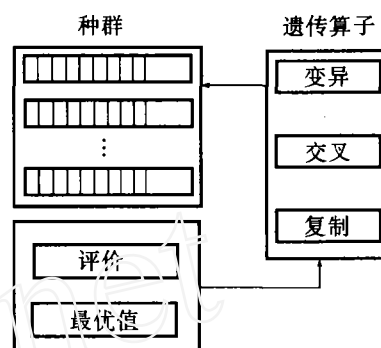


图1 遗传算法流程

遗传算法的特点^[3]: ① 遗传算法直接使用问题参数的适当编码, 而不是参数集本身; ② 遗传算法从问题的解组成的群体开始搜索, 寻找最优解, 而不是从某一点出发; ③ 遗传算法利用输出结果的性能来指导算法朝着最优的方向前进, 而不利用一些特殊的知识. 这样保证了系统的鲁棒性, 可以适用于任何环境; ④ 遗传算法概率地使用3种遗传操作(选择、交叉、变异)进行搜索, 而不用确定的规则, 可以避免传统搜索方法的局部最优的限制, 找到全局最优解; ⑤ 遗传算法具有任意时间算法的特点, 可以在任意时刻中断, 并输出当时的基本较优可行解. 随着规划时间的增加, 系统的性能在增长; ⑥ 遗传算法对待寻优函数基本无限制, 它既不要求函数连续, 也不要求函数可微; ⑦ 遗传算法在解空间进行高效启发式搜索, 而非盲目的穷举或完全随机搜索; ⑧ 遗传算法具有并行计算的特点, 因而可通过大规模并行计算来提高计算速度; ⑨ 遗传算法计算简单, 功能强.

2 GAs toolbox

使用遗传算法需要决定以下几个基本问题: 编码方式、选择函数、遗传算子、初始种群的建立、终止标准以及评价函数等.

2.1 工具箱主要函数清单

主函数: genetic.m; 初始函数: init.m; 选择函数: NGRSel.m rouSel.m touSel.m; 遗传算子: 二进制, sinCro.m binMut.m; 浮点式: heuCro.m ariCro.m bouMut.m uniMut.m NUNut.m MNUMut.m; 终止函数: maxTer.m speTer.m.

2.2 编码方式

在遗传算法中对每一个个体进行适当的编码是必不可少的一步. 算法中问题的构成以及遗传算子的使用都取决于编码方式. Holland的简单遗传算法采用的是二进制编码. 二进制编码理论分析方便, 遗传操作与生物进化相似, 能在相同的范围内表示最多的模式, 能够充分体现隐性的并行性. 但二进制编码也有自己的缺陷, 例如不能直接反映问题的固有结构, 精度不高, 个体长度大, 占用计算机内存多等. Michalewicz对实值编码和二进制编码遗传算法进行了研究比较, 经过实验发现, 使用浮点数编码的遗传算法进行函数优化能取得更好的精度和更高的效率.

2.3 选择函数

选择个体不断地生成下一代在遗传算法中是极其重要的一个环节. 概率选择是根据个体的适配值来进行的, 适配值高的个体被选中的机会也大. 选择策略有: 轮盘赌轮盘选择 rouSel.m(roulette wheel selection), 竞赛选择 touSel.m(tournament), elitist选择, 排序选择(ranking), 几何规划排序选择 NGRSel.m(Normalized geometric ranking).

2.4 遗传算子

遗传算法的基本算子是交叉和变异,算子的应用取决于个体的编码方式.

1) 二进制编码 设 X 和 Y 是 n 维的行向量,用来代表种群中的 2 个父代.采用二进制编码,交叉和变异定义如下:

单点交叉: simSel.m 随机产生一个 1 到 n 之间的数 r ,根据式(1)、(2)产生 2 个新个体 X' 和 Y' :

$$x'_i = \begin{cases} x_i, & \text{if } i < r; \\ y_i, & \text{otherwise.} \end{cases} \quad (1)$$

$$y'_i = \begin{cases} y_i, & \text{if } i < r; \\ x_i, & \text{otherwise.} \end{cases} \quad (2)$$

简单变异: binMut.m 根据式(3)以概率 P_m 对个体的每一位进行改变:

$$x'_i = \begin{cases} 1 - x_i, & \text{if } U(0, 1) < P_m; \\ x_i, & \text{otherwise.} \end{cases} \quad (3)$$

2) 浮点式编码 X, Y 采用实值编码, a_i, b_i 为每一个变量 i 的边界值.

交叉: simCro.m 与二进制式相同,见式(1)、(2).

算术交叉: ariCro.m 当 $r = U(0, 1)$ 时,对父代进行线性算术交叉,见式(4).

$$X' = rX + (1 - r)Y, \quad Y' = (1 - r)X + rY. \quad (4)$$

启发式交叉: heuCro.m 对个体进行线性递推,在 $r = U(0, 1)$ 并且 X 的适配值优于 Y 的适配值的情况下生成新的个体.如果 X' 不可行,那么产生一个新的随机数 r 并重新计算产生一个新的解;否则,停止运算.

统一变异: uniMut.m 随机地选择一个变量 j ,使它等于一个统一的随机数 $U(a_j, b_j)$;边界变异: bouMut.m 随机地选择一个变量 j ,使它等于它的上边界或下边界;非统一变异: NUMut.m 随机地选择一个变量 j ,使它等于一个非统一的随机数;多点非统一变异: MNUMut.m 把非统一变异操作应用于父代 X 中的每一个元素.

2.5 初始函数

init.m 用来提供初始种群,最通常的方法是为整个种群随机地生成解.然而,既然遗传算法自身能够不断地反复提高现有的解,可以在初始种群中加入好的解,剩余的部分再随机产生.

2.6 评价函数

* eval.m 遗传算法与具体应用问题的唯一接口,是种群中个体优劣的一种量化反映;它的构造直接影响求解的效率.许多类型的评价函数都可以用于遗传算法,但它至少要满足一个条件:函数值可以部分有序地分布.评价函数是独立于遗传算法的.

2.7 终止函数

遗传算法从一代到一代通过选择和复制父代直到某一条件被满足.最常用到的终止条件是到某一最大的代数停止(maxTer.m),另一终止策略与种群的收敛标准有关(speTer.m).一般来说,遗传算法将会强迫整个种群的大部分收敛到一个值上.当偏离的个体数量之和小于某一具体极限时,算法就停止.如果在寻优的过程中经过许多代最优值都没有明显改善,当代数累积到一定值时算法也会停止.

3 性能测试^[4]

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

设初始种群大小等于10,采用几何规划排序选择、算术交叉、非统一变异,计算代数分别取1代和25代.实验测试结果如图2、3所示.本算法对单变量的多峰函数寻优十分有效,并且随着计算代数的增加,计算精度得到改善.

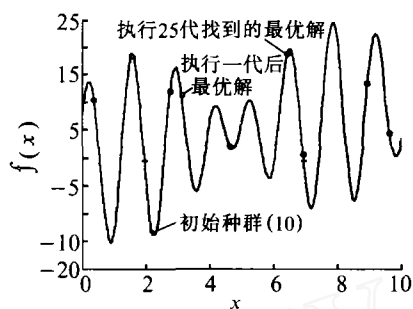


图2 单变量多峰值函数寻优

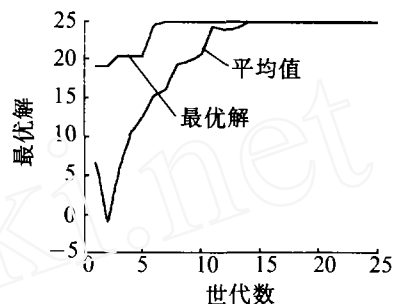


图3 世代函数与精度的关系图

2) 测试函数

$$f(x_1, x_2) = x_1 \sin 3\pi x_1 + \sin 12\pi x_2;$$

$$x_1 = [-3, 10], x_2 = [4, 6].$$

对测试函数分别使用二进制编码和浮点式编码进行比较.实验结果如图4、5所示(初始种群取80,执行100代),浮点式在求解效率和稳定性上优于二进制遗传算法.

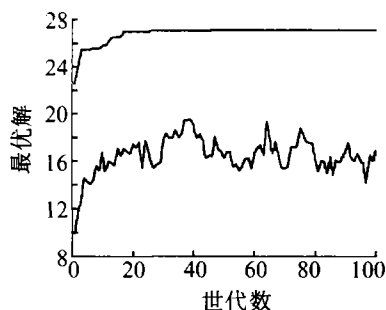


图4 二进制遗传算法

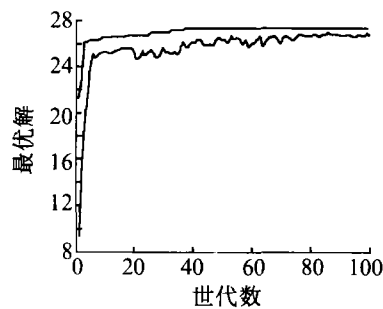


图5 浮点式遗传算法

3) Corana 函数寻优

Corana 函数是一系列参数变量函数 q_n , 非常便于计算并且含有许多的局部最小值. Corana 函数是一个直角口袋状 n 维空间抛物线(见图6),它的全局最小值产生在 $(0, 0, 0, \dots)$. 使用不同的随机种子对测试函数分别用二进制(BGA)和浮点式(FGA)遗传算法测试10次,并且与使用模拟退火算法(SA)求得的结果进行比较,结果见表1.

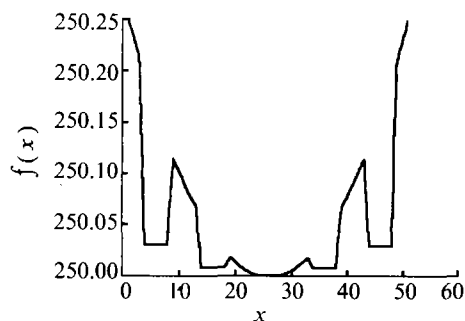


图6 平面 Corana函数

表1 算法比较

维数	方法	平均值	最小值
2	FGA	3.82e-8	2.03e-8
	BGA	3.41e-7	2.16e-8
	SA	1.13e-8	4.21e-10
4	FGA	5.60e-7	1.37e-7
	BGA	5.254e-7	2.91e-9
	SA	6.18e-4	8.70e-9
10	FGA	5.64e-7	1.49e-8
	BGA	0.62e+2	0.13e+1
	SA	5.40e-4	5.40e-4

实验结果显示:在低维数的情况下,算法之间区别不大;到了高维数的问题,FGA 明显优于其他两种算法。FGA 在 3 种维数情况下都能够找到最优点;SA 在四维情况下,有 2 次没有找到最优解;在十维情况下,根本无法找到最优解。BGA 在低维数情况下有较好的表现;到了高维数,寻解效率明显降低。

4 结 语

完成 Matlab 遗传算法工具箱的设计,支持二进制和浮点式编码方式,工具箱提供了一个标准化的、可扩展的、简单的算法,广泛适用于各种复杂的数值运算。工具箱包括了大量的算子函数,提供各种类型的选择策略,交叉、变异的方式,适用于各类不同的问题。使用工具箱对一系列非线性、非凸、多峰值函数进行了测试,取得了令人满意的结果。

遗传算法的控制参数对算法本身有非常重要的影响,对遗传算子的适当改进能够大大改善算法的性能。通过实验可知,对于非线性、多峰值函数的寻优问题,浮点式遗传算法(FGA)与传统遗传算法(BGA)以及传统寻优方法相比,具有明显的优势。它不会陷入局部最优,有较强的搜索能力,而且收敛速度快、精度高。

遗传算法作为一种解决复杂问题的有效的优化方法,近些年来越来越受到人们的重视。它已经在机器学习、软件技术、图像处理、模式识别、神经网络、工业优化控制、生物学、遗传学、社会科学等方面有了不少应用。目前,遗传算法正向其他学科和领域渗透,正在形成遗传算法、神经网络和模糊控制相结合,从而构成了一种新型的智能控制系统整体优化的结构形式。

参考文献:

- [1] 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.
- [2] 张晓绩,方 浩,戴冠中.遗传算法的编码机制研究[J].信息与控制,1997,2: 35-39.
- [3] MICHALEWICZ Z. Genetic algorithm+Data structures=Evolution programs[M]. AI Series. New York: Springer-Verlag, 1994.
- [4] CORANA A, MARCHESI M, MARTINI C, et al. Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm [J]. ACM Trans math software, 1987, 13(3): 262-280.

Design of GAs Toolbox Matlab

GUO Qiang, ZHAO Xiao-hua, SUN Liang

(College of Electronic Information and Control Engineering, Beijing Polytechnic University, Beijing 100022, China)

Abstract: Owing to adopting the Matlab language, the genetic algorithms (GAs) are achieved and further, GAs Toolbox was set up. The developed GAs Toolbox supports both the binary and the floating-point encoding methods so that the performance of GAs is enhanced. The toolbox was used to test a series of nonlinear, multi-peak-value functions and demonstrates excellent performances. It is also found that the floating-point GA is superior to the binary GA and the traditional optimization methods in solution-finding rate and exactness.

Key words: genetic algorithms(GAs); function optimization; Matlab