广西科学院学报　　2007,23(2):70～72
Journal of Guangxi Academy of Sciences　　Vol. 23, No. 2　May 2007

求分圆多项式近似根的遗传算法*

Genetic Algorithm for Finding Approximate Roots of a Cyclotomic Polynomial

刘向虎，何登旭
LIU Xiang-hu, HE Deng-xu

(广西民族大学数学与计算机科学学院，广西南宁　530006)
(College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning, Guangxi, 530006, China)

摘要：新提出的求分圆多项式近似根的遗传算法，是取 m 个个体，在初始群体中随机产生 m 个初始点，再用适应度函数 $1/(1+|f(x)|)$ 计算个体适应度，对种群进行选择、交叉、变异操作，将适应度好的个体组成下一代群体，直到达到规定近似根的个数和精度，就输出结果. 该算法采用动态自适应技术、重新启动法、多项式除法等措施进行优化，可以有效地防止出现未成熟收敛问题. 该算法在求分圆多项式的近似根方面是可行的，并取得比较好的效果，为判定一个多项式是否分圆提供了一种新方法.

关键词：遗传算法　分圆多项式　根
中图法分类号：TP183　**文献标识码**：A　**文章编号**：1002-7378(2007)02-0070-03

Abstract: The newly-proposed genetic algorithm for finding approximate roots of a cyclotomic polynomial in this paper refers to the process in which m individuals are selected to produce randomly m original points in the original groups. Then by using the fitness function $1/(1+|f(x)|)$, individuals' fitness is calculated, selection, crossover and mutation of initial population are performed to integrate individuals with good fitness into the next generation. The result will come out when the ordered number of the approximate roots and the precision are achieved. The algorithm is optimized by using dynamic adaptive, restartion, polynomial division to restrain the premature convergence efficiently. The algorithm is practicable and efficient in finding approximate roots of a cyclotomic polynomial, and gives a new method to judge a polynomial that can pitch cirle.

Key words: genetic algorithm, cyclotomic polynomial, root

遗传算法(Genetic Algorithm, GA)是由美国密执安大学的 Holland 教授首先提出，后经过 DeJong、Goldberg 等人归纳总结所形成的一类模拟进化算法[1]. 遗传算法是借鉴生物界自然选择和群体进化机制而形成的一种全局寻优算法，其主要优点是简单、通用、鲁棒性强，适合并行分布处理[2]. 遗传算法以其很强的解决问题的能力和广泛的适应性已经渗透到研究与工程的各个领域，特别是在机器学习、智能控制、组合优化等方面，遗传算法已经得到了广泛应用，并取得了良好的结果[3].

在多项式理论和应用中，分圆多项式是一类比较特殊的多项式，它有许多特殊的性质，如分圆方程是正规等[4]. 由于分圆多项式在编码理论中的重要应用，对它的研究就显得更加必要[5]. 本文提出一种求分圆多项式近似根的遗传算法，实例应用结果表明，该算法在求分圆多项式的近似根方面是可行的，并取得比较好的效果.

1　遗传算法的一般解题步骤[3]

遗传算法的一般解题流程如图 1 所示.

收稿日期：2006-11-09
修回日期：2006-12-14
作者简介：刘向虎(1980-)，男，硕士研究生，主要从事智能优化算法方面的研究工作。
* 国家自然科学基金(60461001)，国家民委科学基金(05GX06)资助项目。

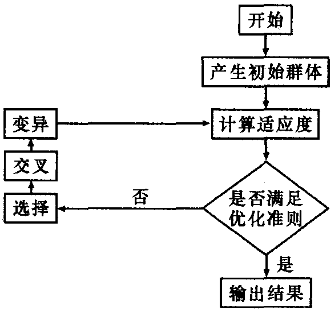

图 1 遗传算法的一般解题流程

算法步骤如下.

步骤 1:随机产生初始群体.

步骤 2:计算各个体的适应度.

步骤 3:根据遗传概率，用下列操作产生新群体.

(1)复制.将已有的优良个体复制后添入新群体中,并删除劣质个体.

(2)交换.将选出的 2 个个体进行交换，所产生的新个体添入新群体.

(3)突变.随机的改变某个体的某个基因后,将新个体添入新群体.

步骤 4:反复执行步骤 2 和步骤 3,直到满足终止条件后,选择最佳个体作为结果输出.

2 求分圆多项式近似根的遗传算法描述

2.1 产生初始群体

取 m 个个体,初始群体中的每一个个体都是按随机方法产生的，这样可以保证搜索的广度.由于分圆多项式的近似根是分布在一个单位圆上的,所以可以根据复数来设计[6],设近似根 $x=e^{i\theta}$,$\theta=2\pi r$,用二进制对 r 进行编码,在这里取 47 位二进制数,因为在 Matlab 中用 format long 语句可以达到的小数位数是 14 位,转化为整数后,再转化为二进制数的位数是 47 位[7].设 $a=47$,b 为随机整数($b<2^a$),选取 m 次,那么设 $r=b/2^a$，显然 $r\in[0,1)$,这样就得到了随机产生的 m 个 r，也就得到了随机产生的 m 个初始点.

2.2 计算个体适应度

取适应度函数 $1/(1+|f(x)|)$ 计算个体适应度,其中 $|x|=1$($|x|$ 表示复变量 x 的模),$f(x)$ 为分圆多项式.

2.3 对种群操作

(1) 选择操作.在遗传算法中通过复制，将选择出来的优良个体插入下一代新群体，体现“优胜劣汰”的原则.即对种群中所有的个体的适应度进行计算，如果适应度大于规定的精度，则记录下来，插入下一代群体，反之，则往下进行.假如已经选出了 p 个个体，在整个群体中再任意选择 q 个个体($0<q\leqslant m$),对这 q 个个体,分别计算它们的适应度,选择一个最优个体,重复 $m-p$ 次,加上已经选出的 p 个个体,就产生了下一代群体.这一策略显然能够保证整个种群中解的最高适应值在迭代过程中不会下降，能确保优秀的基因能够继续遗传和繁衍.

(2) 交叉操作.采用均匀交换的原则,即对选择出的 2 个父代个体的每一个基因都以相同的交叉概率进行交换，从而形成两个新的个体.具体过程如下：

a.随机产生一个与群体中编码长度相同的个体,设它的编码是 $w=m_1m_2\cdots\cdots m_1$.

b.随机选取两个父代个体 A_1、B_1,产生的个体是 A_2、B_2,如果 $m_i=0$,则 A_2 的第 i 个基因继承 A_i 对应的基因,B_2 的第 i 个基因继承 B_1 对应的基因.如果 $m_i=1$,则 A_2 的第 i 个基因继承 B_1 对应的基因,第 i 个基因继承 A_i 对应的基因.

(3)变异操作.对某一个字符进行补运算,即将字符 1 变成 0,或将 0 变成 1.

2.4 下一代群体的形成

将适应度好的个体组成下一代群体,再重复进行 2.2～2.3.

2.5 终止条件

达到规定的近似根的个数和精度就可以停止，此时的结果就是近似解了.对于迭代次数和精度来说,如果提前达到规定的精度,则停止,输出结果.

3 算法优化

在实际的应用中,遗传算法显示出顽强的生命力，以其独特的优点越来越受到人们的重视.但是,遗传算法也存在着缺点,在实际的应用中也因此出现了一些问题[3],其中很重要的是未成熟收敛(也称早熟),即当还未达到全局最优解或满意解时，群体中不再产生性能超过父代的后代，群体中的个体十分的相似.未成熟收敛的主要特征是群体中结构的多样性急剧的减少.这将导致遗传算法的选择操作和交叉操作不再产生更有生命力的新个体.遗传算法希望得到最优解或满意解，而不是在找到最优解或满意解之前，整个群体收敛到一个非优个体，它希望能够保持群体中结构的多样性，从而使搜索进

行下去. 在本文的算法中也碰到了同样的问题,为了防止上述问题的出现,我们采取以下 3 个措施:

(1) 采用动态自适应技术,即将交叉概率和变异概率动态化. 这里设 t 为求出的根的个数,s 为最高次幂,那么交叉概率可以设为 $pc = 1 - t/s$,变异概率为 $pm = t/s$,可以看出算法的初期交叉概率大,可以从全局搜索,加快收敛速度;后期变异概率大,扩大搜索范围,增加群体的多样性,防止未成熟收敛.

(2) 采用重新启动法. 即当遗传算法搜索中碰到未成熟收敛问题时,则随机一组初始值重新进行遗传操作.

(3) 多项式除法. 当出现给定多项式的一个根时,用原多项式将其除去,这样可以防止重根的出现.

4 实例应用

我们在联想微机上采用 Matlab7. 04 软件实现本文算法. 分别用本文提出的求分圆多项式近似根的遗传算法和 Matlab7. 04 直接求根法,在微机上求分圆多项式函数 $f(x) = x^4 - x^3 + x^2 - x + 1$ 与 $f(x) = x^{16} + x^{14} - x^{10} - x^8 - x^6 + x^2 + 1$ 的根.

微机的基本配置是 pentium(R) 4,cpu 2. 4GHZ,256M 的内存. 取初始种群 $m = 200$,迭代次数 $nga = 50$,精度 $\varepsilon = 10^{-10}$,其最高次幂既是要求的近似根的个数,结果见表 1 和表 2.

表 1 求多项式函数 $f(x)=x^4-x^3+x^2-x+1$ 的根的结果

迭代次数	本文算法求得的根	Matlab7. 04 求得的根	精度
16	0. 80901699437497− 0. 58778525229244i	0. 80901699437495− 0. 58778525229247i	1. 000000 00000 00e−013
16	0. 80901699437494+ 0. 58778525229248i	0. 80901699437495+ 0. 58778525229247i	
16	−0. 30901699437497− 0. 95105651629515i	−0. 30901699437495− 0. 95105651629515i	
16	−0. 30901699437489+ 0. 95105651629517i	−0. 3090169943749+ 0. 95105651629515i	

表 1 和表 2 的结果表明,本文算法在求分圆多项式方面是可行、有效的,并具有较高的计算精度和较少的迭代次数.

5 结束语

文中给出求解分圆多项式近似根的遗传算法,通过实例可以看出,该算法在求分圆多项式的近似根方面是可行、有效的,并具有较高的计算精度和较少的迭代次数,它还为判定一个多项式是否分圆提

表 2 求多项式函数 $f(x)=x^{16}+x^{14}-x^{10}-x^8-x^6+x^2+1$ 的根的结果

迭代次数	本文算法求得的根	Matlab7. 04 求得的根	精度
13	− 0. 20791169085836 + 0. 97814760070171i	− 0. 20791169081776 + 0. 9781476007338i	2. 000000 000000 00e − 10
13	− 0. 20791169084162 − 0. 97814760074497i	− 0. 20791169081776 − 0. 9781476007338i	
13	0. 20791169087823 + 0. 97814760072095i	0. 20791169081776 + 0. 9781476007338i	
13	0. 20791169082740 − 0. 97814760073176i	0. 20791169081776 − 0. 9781476007338i	
13	0. 99452189536620 + 0. 10452846328741i	0. 99452189536827 + 0. 10452846326765i	
13	0. 99452189536705 − 0. 10452846327932i	0. 99452189536827 − 0. 10452846326765i	
13	− 0. 99452189536669 − 0. 10452846328274i	− 0. 99452189536827 − 0. 10452846326765i	
13	− 0. 99452189538369 + 0. 10452846312095i	− 0. 99452189536827 + 0. 10452846326765i	
13	0. 40673664302243 + 0. 91354545766636i	0. 40673664307580 + 0. 91354545764260i	
13	0. 40673664310555 − 0. 91354545762936i	0. 40673664307580 − 0. 91354545764260i	
13	− 0. 40673664308373 − 0. 913545457643907i	− 0. 40673664307580 − 0. 91354545764260i	
13	− 0. 40673664308283 + 0. 91354545763947i	− 0. 40673664307580 + 0. 91354545764260i	
13	0. 74314482553520 − 0. 66913060629466i	0. 74314482547739 − 0. 66913060635886i	
13	0. 743144825476007 − 0. 66913060637810i	0. 74314482547739 − 0. 66913060635886i	
13	− 0. 74314482548620 + 0. 66913060634907i	− 0. 74314482547739 + 0. 66913060635886i	
13	− 0. 74314482548196 − 0. 66913060635378i	− 0. 74314482547739 − 0. 66913060635886i	

供了一种方法. 基于区间分布的多项式根的研究和有关的工作我们将在下一步进行.

参考文献:

[1] 周明,孙树东. 遗传算法原理及应用[M]. 北京:国防工业出版社,2002.

[2] 熊伟清,魏平,赵杰煜. 遗传算法的早熟现象研究[J]. 计算机应用研究,2001(9):12-14.

[3] 雷英杰,张善文,李续武,等. MATLAB 遗传算法工具箱及应用[M]. 西安:西安电子科技大学出版社,2005.

[4] 刘国华,包宏,李文超. 用 MATLAB 实现遗传算法程序[J]. 计算机应用研究,2001(8):80-82.

[5] 周永权,马武瑜. 基于代数神经网络的分圆多项式判定及学习算法[J]. 系统工程与电子技术,2000,22(5):79-82.

[6] 程锦松,刘锋. 基于分布理论和遗传算法的多项式求根算法[J]. 微机发展,2001(6):1-2.

[7] 王沫然. MATLAB 与科学计算[M]. 第 2 版. 北京:电子工业出版社,2004.

(责任编辑:凌汉恩 邓大玉)