非线性边值问题中的遗传算法

Genetic Algorithms in the Study of Nonlinear Boundary Value Problems

刘希玉

(山东师范大学信息管理学院　济南250014)

Abstract In this paper, a totally new approach of finding approximate solutions to nonlinear boundary value problems is presented. In contrast to the traditional way iteration, we apply the genetic algorithms in the field of artificial intelligence. Experiments show that for superlinear and sublinear boundary value problems, the approximate solutions converge satisfactorily. Visualisation of the process and error line figures is also given.

Keywords Genetic algorithms, Nonlinear problems, Boundary value problems, Visualisation

1. 引言

遗传算法是基于自然选择的一种搜索算法，它的依据为达尔文所发现的适者生存的进化规律。自从该算法提出以来，遗传算法在优化理论、数值分析等领域得到了大量有效的应用，有关的参考文献可见文[4]。

在微分方程领域，迭代方法被广泛使用。在文[1～2]中，Lakshmikantham，Heikkila，Ladde，及 Vatsala 系统研究了单调迭代方法在微分方程的边值问题中的应用，文[3]中还研究了迭代方法在高阶问题中的应用。然而，在许多问题中，由于确定上下解是个困难的问题，因此，用迭代方法来求微分方程的数值解并不总是有效的方法。

另一方面，虽然遗传算法在优化理论中得到了有效的应用，在微分方程边值问题中还没有发现其相应的算法及应用。本文的目的为将遗传算法引入到上述问题中。我们给出了解决边值问题的算法以及 C++ 实现。系统采用 Visual C++ 及 MatLab 完成了求解过程的可视化。针对6类非线性问题，我们的系统显示遗传算法在这类问题的解决中是有效的。我们还同时给出了错误曲线。

设 $p\in C^1(0,1)$为连续且为正值。考虑下列问题：

$$\begin{cases} -Lx=f(t,x(t)),t\in(0,1) \\ \alpha x(0)-\beta\lim\limits_{t\to 0}p(t)x'(t)=0 \\ \gamma x(1)+\delta\lim\limits_{t\to 1}p(t)x'(t)=0 \end{cases} \tag{1.1}$$

其中$(Lx)(t)=(p(t)x'(t))'/p(t)$。假设 $p(t)$满足(见文[5])

$$\int_0^1\frac{dt}{p(t)}<\infty \tag{1.2}$$

函数 $f(t,x)$称为问题的非线性项。一般情况下，我们将假设 $f:[0,1]\times R^1\to R^1$连续，且 $\alpha,\beta,\gamma,\delta\geqslant 0,\alpha\delta+\beta\gamma+\alpha\gamma>0$。记

$$\tau_0(t)=\int_0^t\frac{dt}{p(t)},\tau_1(t)=\int_t^1\frac{dt}{p(t)},$$

称 $x\in C[0,1]\cap C^2(0,1)$ 为问题(1.1)的解，如果 $x(t)$ 满足(1.1)及边界条件。

有关问题(1.1)的存在性结果有很多，参见文[5～7]。本文的主要目的是给出一个构造性的方法。另外我们给出了收敛性分析。

2. 遗传算法

遗传算法是一种搜索算法，其依据为达尔文的适者生存原则，即具有最大适应度的个体得到生存。近年来，遗传算法以及进化技巧已经广泛应用于人工智能、优化理论等学科，得到了若干深刻的结果。

遗传算法用个体组成的集合一群体(population)来工作。所有个体作为潜在的问题的解。个体由二进制的字符串组成，也称作染色体。个体通过若干变换使得群体得以进化。进化的原则为保留适应度大的个体。当群体达到一个稳定状态时，个体就是所要的最优解。近年来，David Goldberg 对遗传算法又作了进一步的扩展。

遗传算法与若干优化算法如最速下降法、人工神经网络等具有明显的不同，这表现在：

(1)遗传算法的工作个体为参数的编码而不是参数本身；

(2)遗传算法同时在一个群体中寻找潜在的解而不是处理一个个体；

(3)遗传算法使用概率法则(如变异)而非确定性法则；

(4)遗传算法遍历一个可能解的数据库，因此它能得到全局最优解，一般不会陷入局部极小。

最近，遗传算法在空气动力学以及机翼设计中得到了有效的应用。其优点为算法简单，并能处理非线性及非连续的情况。另外，遗传算法能够同时处理混合多参数，如浮点型、整型、逻辑型并存的问题。遗传算法的另一个特点为它可以解决多目标规划问题，在这种下，有多种不同的约束存在，从而解是一个集合而非单个的解。

应用遗传算法的关键是找到一个合适的适应度函数，同时又不至于产生不可接受的计算量。一般来说，复杂的问题常导致大的群体和长的染色体。但是，遗传算法可以有效地利用系统的并行计算功能，这是因为在一个进化代中，对个体的适应度评估可以并行进行。

遗传算法的基本过程如下：

(1)初始化初始群体；

(2)根据适应度函数计算个体的适应度并选择具有高适应度的个体；

(3)应用遗传算子(交叉、变异)得到新的群体，然后回到(2)，直到得到一个更好的解。

选择运算是一个类似锦标赛的过程。首先我们随机选择两个个体，然后根据其适应度选取适应度大者。所得到的个体进入下一代。由于这个过程不能产生全新的个体，因此需要有交叉算子来进行新的运算。交叉有两步，首先随即选择两个个体作为父代，其次两个父代中的个体进行交叉运算，即随机选择一个整数 k 位于区间$[1,l-1]$中，其中 l 为染色体的长度，

然后交换它们串中的 $k+1$至 l 部分。

变异算子旨在避免交叉有时可能丢失有益的遗传基因。在二进制编码的情况，变异为按照某种概率将其串中的1或0翻转。

3. 问题的描述

为利用遗传算法讨论问题(1.1)，我们需要将其做以下变换。首先记

$$\rho^2 = \beta\gamma + \alpha\delta + \alpha\gamma\int_0^1 \frac{1}{p(t)}dt$$

容易知道 $\rho > 0$。设

$$u(t) = \frac{1}{\rho}[\delta + \gamma\tau_1(t)]$$

$$v(t) = \frac{1}{\rho}[\beta + \alpha\tau_0(t)]$$

因此 $\gamma v + \alpha u = \rho$。定义格林函数如下(见文[5])：

$$G(t,x) = \begin{cases} u(t)v(s)p(s), 0 \leqslant s \leqslant t \leqslant 1 \\ v(t)u(s)p(s), 0 \leqslant t \leqslant s \leqslant 1 \end{cases} \quad (3.1)$$

根据假设(1.2)我们知道格林函数连续。因此问题(1.1)等价于下列积分方程：

$$x = Ax = \int_0 G(t,s)f(s,x(s))ds, x \in C[0,1]$$

定义线性算子 G

$$(Gx)(t) = \int_0 G(t,s)x(s)ds, x \in L^\infty[0,1] \quad (3.3)$$

引理3.1 设 $e(t)$为下列问题的解

$$\begin{cases} -Lx = 1, t \in (0,1) \\ \alpha x(0) - \beta \lim\limits_{t\to 0} p(t)x'(t) = 0 \\ \gamma x(1) + \delta \lim\limits_{t\to 1} p(t)x'(t) = 0 \end{cases} \quad (3.3)$$

则

$$e(t) = \frac{1}{\rho^2}(\delta + \gamma\tau_1(t))\int_0[\beta + \alpha\tau_0(s)]p(s)ds + \frac{1}{\rho^2}(\beta + \alpha\tau_0(t))\int[\delta + \gamma\tau_1(s)]p(s)ds$$

证明：由格林的定义直接计算即可。

现在我们描述工作空间。首先定义一个权函数如下：

(1) $\theta(t) = \tau_1(t)$若 $\beta > 0, \delta = 0$；

(2) $\theta(t) = \tau_0$若 $\beta = 0, \delta > 0$；

(3) $\theta(t) = \tau_0(t)(t)$，当 $t \in [0,1/2]$；$\theta(t) = \tau_1(t)$，当 $t \in [1/2,1]$；若 $\beta = 0, \delta = 0$。

(4) $\theta(t) = 1 + \tau_1(t)$若 $\beta, \delta, \gamma > 0, \alpha = 0$；

(5) $\theta(t) = 1 + \tau_0(t)$若 $\beta, \delta, \alpha > 0, \gamma = 0$；

(6) $\theta(t) = 1 + \tau_0(t)$，当 $t \in [0,1/2]$；$\theta(t) = 1 + \tau_1(t)$，当 $t \in [1/2,1]$若 $\alpha, \beta, \gamma, \delta > 0$。

定义巴拿赫空间如下

$$E = \{x : x \in C[0,1], \sup_{t\in(0,1)} \frac{|x(t)|}{\theta(t)} < \infty\}$$

范数定义为

$$\|x\| = \sup_{t\in(0,1)} \frac{|x(t)|}{\theta(t)}$$

直接计算可知下列估计成立：

引理3.2 存在常数 $C > 0$满足

$$0 \leqslant G(t,s) \leqslant C\theta(s)p(s), t,s \in (0,1)$$

有上述引理容易知道算子 A 是紧的。函数 e(t)的作用相当于基本解，用于构造交叉和变异算子。

4. 算法描述

本节我们描述用于问题(1.1)的遗传算法。算法包括以下一些部分：初始化；编码方法；遗传算子；选择；适应度函数。

·编码

首先我们给出问题解(个体)的编码方法。基本思想为离散化。设 x 为 E 中的一个个体，N 为自然数。我们将把 x(t)表示为一个长度为 N 的 double 型数组，其元素为 $x(t_i), t_i = i/N$。如下代码片断说明以上方法：

```
#define N 100
double * m_pData=new double[N];
double * m_p1Diff=new double[N];
double * m_p2Diff=new double[N];
m_pData=NULL;
m_p1Diff=NULL;
m_p2Diff=NULL;
```

这里我们使用了稍微多一点的数据结构，其目的是保证连续性和光滑性。指针 m_p1Diff 和 m_p2Diff 为函数 $x(t) \in E$ 的一阶和二阶导数。因此，一个个体的光滑构造为

$$x(t) = x(0) + x'(0)t + \frac{1}{2}x''(0)t^2 + \frac{1}{2}\int_0(t-s)^2h(s)ds$$

其中 h(s)为有界函数。上述函数将具有连续二阶导数。

·磨光算子

由于交叉可能产生不连续性，因此我们需要一个磨光算子用于处理光滑化的问题。磨光算子为一个非负、光滑的函数 $J \in C_0^\infty(R)$满足

(i) $J(x) = 0$ *if* $|x| \geqslant 1$，

(ii) $\int J(x)ds = 1$。

例如，下列函数为一个磨光函数

$$J(x) = \begin{cases} ke^{-1/(1-|x|^2)}, & |x| < 1 \\ 0, & |x| \geqslant 1 \end{cases}$$

其中

$$\frac{1}{k} = \int_{|x|<1} e^{-1/(1-|x|^2)}dx$$

取 $\varepsilon > 0, J_\varepsilon(x) = \varepsilon^{-1}J(x/\varepsilon)$。定义卷积

$$J_\varepsilon * u(x) = \int J_\varepsilon(x-y)u(y)dy$$

称为 u 的磨光算子。注意这里函数 u(x)可以不连续。由于本文的需要，我们定义如下另一种磨光函数：

$$J(x) = \begin{cases} 1/2, & |x| < 1 \\ 0, & |x| \geqslant 1 \end{cases}$$

其计算量较小，而且对给定的有界函数，其磨光连续。

·交叉和变异

设 x(t)，y(t)为空间 E 中的两个解，交叉算子定义为下述变换：

$$E \times E \to E \times E : (x,y)a(u,v)$$

$$u = GJ_\varepsilon * u^h, v = GJ_\varepsilon * v^h$$

$$u^h = \begin{cases} x(t), t \in [0,1]\backslash(t_0,t^0) \\ y(t), t \in (t_0,t^0) \end{cases}$$

$$v^h = \begin{cases} y(t), t \in [0,1]\backslash(t_0,t^0) \\ x(t), t \in (t_0,t^0) \end{cases}$$

其中算子 G 由(3.3)式定义，$\varepsilon > 0, 0 \leqslant t_0 \leqslant t^0 \leqslant 1$为变换点。变换点的选择是随机的。粗略来说，交叉算子在一个随机的区间上交换函数的一部分。为使得交叉后的函数光滑，我们使用了磨光算子。最后的映射通过算子 G 来实现，目的为保证边界条件的满足。

下面描述变异算子。设 $P \in (0,1)$为一个小概率，变异定义为

$$L^\infty[0,1] \to E : ua\ x, x = Gu^h$$

$$u^h = c_0 + c_1 + \frac{1}{2}c_2t^2 + 12\int_0(t-s)^2u(s)ds$$

其中 c_0, c_1, c_2为常数，u 为以概率 P 选择的函数，满足下面的

限制条件。定义函数 u 在点 t 的变差为

$$\omega(u,t)=\inf_{a>0}\left(\operatorname*{esssup}_{s\in(t-a,t+a)}u(s)-\operatorname*{essinf}_{s\in(t-a,t+a)}u(s)\right)$$

我们将限制选取的函数 u 满足

$$\omega(u,t)<\omega_0,t\in(0,1)$$

第二个变异算子的定义如下：$x=Gu^k$ 其中 $u^k=GJ_{\cdot}\cdot u$。

·适应度函数和选择

为进行选择运算，我们需要定义适应度函数。我们定义两种函数，分别对应于 L_2 和 L^∞ 范数。设 $x\in E$，定义函数 Fit1和 Fit2如下：

$$\text{Fit1}:E\to R^1,Fit1(x)=\left(\int_0^1|x-Ax|^2\right)^{1/2}$$

$$\text{Fit2}:E\to R^1,Fit2(x)=\sup_{t\in(0,1)}|x(t)-Ax(t)|$$

$$\text{Fit3}:E\to R^1,Fit3(x)=\|x-Ax\|$$

相应地，我们称它们为平均误差、最大误差以及标准误差。

选择将基于群体中的一个比较和排序过程。为此，我们在空间 E 中定义一个伪序为 p，其中适应度函数可以为上述的任何一种：

$x\leqslant y\Leftrightarrow x\ \mathrm{p}\ y$ 且

$P(Fit(y)-Fit(x)>er_0)<p_0$，

$Fit(y)-Fit(x)\leqslant er_1$

这里 x，y 为任意个体，p_0为一个小概率，$0<er_0<er_1$为常数。

关系可以用来保证收敛性，见第5节。在一些问题中，为简化程序，也可以不使用这个关系。

最后，选择为：

(i)根据伪序 p 对群体进行排序；

(ii)根据特定的整数 N_0丢弃群体中的最后部分；

(ii)根据变异运算在群体中增加 N_0新个体。

·群体和解

设群体(population)的大小为 N。初始群体由变异运算得到，也就是说，具有一定随机性的函数。在进化过程中，交叉和变异使得群体一代一代地进化，最终得到解。

算法终止的条件为满足给定的精度要求。此时，群体中第一个个体就是所要的解。当然，这是一个数值解。

5. 收敛性分析

本节我们给出次线性情况下的收敛性分析。设函数 f(t，x)连续，且为次线性，也就是说，下列条件满足：

(F1)$\lim_{x\to\infty}\frac{f(t,x)}{x}=0$ 对 $t\in(0,1)$一致成立。

(F2)对 $t\in(0,1)$，$x\in R^1$，函数 x-f(t，x)严格增。

在上述条件下，本文第4节中的算法收敛。

引理5.1 设条件(F1)满足，则问题(1.1)可解。

证明：根据条件(F1)及标准的不动点论证方法我们知道算子 A 存在不动点 $x^*\in E$。因此问题(1.1)可解。

为了得到比较定理，我们考虑下列线性方程：

$$\begin{cases}-Lx=x,t\in(0,1)\\ \alpha x(0)-\beta\lim_{t\to0}p(t)x'(t)=0\\ \gamma x(1)+\delta\lim_{t\to1}p(t)x'(t)=0\end{cases}\tag{5.1}$$

其对应的算子方程为

$$x=Gx,x\in E\tag{5.2}$$

定理5.2 设条件(F1)(F2)满足，$\Omega\subset E$ 为有界开区域使得问题(1.1)在 Ω 有唯一解。设 $x_1,x_2\in\Omega$，x^* 为问题(1.1)在中的解。进一步假设 x_1-x_2为线性问题(5.2)的下解。则对 $t\in(0,1)$下列估计成立

$$x_1(t)-x^*(t)<x_2(t)-x^*(t)\Rightarrow$$

$$x_1(t)-(Ax_1)(t)\leqslant x_2(t)-(Ax_2)(t)\tag{5.3}$$

$$x_1(t)-x^*(t)>x_2(t)-x^*(t)\Rightarrow$$

$$x_1(t)-(Ax_1)(t)\geqslant x_2(t)-(Ax_2)(t)\tag{5.4}$$

证明：由于(5.3)和(5.4)的证明类似，故我们只证明(5.3)。设 $y_1=Ax_1,y_2=Ax_2,u_1=v_1-y_1,u_2=v_2-y_2$其中

$$v_1=\int_0^1G(t,s)x_1(s)ds,v_2=\int_0^1G(t,s)x_2(s)ds$$

容易知道 u_1，u_2满足下列方程

$$\begin{cases}-Lx=x(t)-f(t,x(t)),t\in(0,1)\\ \alpha x(0)-\beta\lim_{t\to0}p(t)x'(t)=0\\ \gamma x(1)+\delta\lim_{t\to1}p(t)x'(t)=0\end{cases}\tag{5.5}$$

其中 $x=x_1$或 $x=x_2$，相对应于 u_1或 u_2。由条件(F2)以及(5.3)式中的假设我们得到下列估计

$$x_1(t)-f(t,x_1(t))\leqslant x_2(t)-f(t,x_2(t))\tag{5.6}$$

因此由(5.5)式及标准比较估计得到 $u_1(t)\leqslant u_2(t)$。证毕。

从以上分析我们看到，一个更好的解意味着从 x 到 Ax 更小的距离。因此我们可以用 x 到 Ax 的距离来刻画个体的适应度。

命题5.3 设(F1)(F2)满足，$\Omega\subset E$ 为有界开区域使得问题(1.1)在 Ω 中有唯一解。设 $x,y\in\Omega$，定义关系 P

$x\ \mathrm{p}\ y\Leftrightarrow x-y$ 为(5.2)的下解

在下述意义下第4节的算法收敛，即只要解在 Ω 中，则误差减小。

6. 系统实现

本节我们描述系统实现。所用工具为 Visual C++6.0，可视化引擎为 MatLab version 6.1。主要的数据结构如下的类 CSolution。

Class CSolution

```
// Solution.h: interface for the CSolution class.
//////////////////
Class of a solution function in the interval [0,1].
//////The slicing interval is 0.01, the size of data is 101.
#include "math.h"
class CSolution
{
public:
    CSolution();
    virtual $\sim$CSolution();
//data members
private:
    double * m_pData;
    double * m_p1Diff;
    double * m_p2Diff;
    //member functions
    public:
    void Smooth(int nScope=5);
    void SetData(int nPos, double data);
    void SetData(double * ptData);
    void SetData(CSolution * pSolu);
    double GetValue(int nPos);
    void Create2Smooth();
    void Create1Smooth();
    void Compute2Diff();
    void Synchronize();
    void Compute1Diff();
    void GetData(double * m_pTarget);
    CString GetStrData();
    void CreateRandom();
    void CreateContinuous();
    double GetRand(double fMin=-10,double fMax=10);
    CString GetStrRand(double fMin=10, double fMax=10);
    CString GetStr1Diff();
    CString GetStr2Diff();
};
```

交叉、变异、选择等函数的定义如下：

Main GA functions

```
//ga functions
public :
        void Cross(CSolution \&soluA, CSolution \&soluB);
        void Cross(int nNumber);
        void Mutation(int nNumber=5);
        void SortSolu();
        void CalcuErr(CSolution \&solu);
        double Get1.2Err(CSolution \&solu);
        double GetMaxErr(CSolution \&solu);
}
```

实验数据采用下列6种非线性项：

(1)超线性指数函数(SEN)。

$$f(t,x)=x^2\sin(t)\sin(x)+e^t\sin(x+3)+te^t+11$$

(2)次线性函数(SF)。

$$f(t,x)=x\sin(t)+5+\sqrt{|x|}$$

(3)超线性多项式(SP)。

$$f(t,x)=4x^4-2\sin(t)x^3+7x^2+xe^t-5$$

(4)弱奇性函数(WSN)。

$$f(t,x)=4x-\frac{2\sin(t)}{1-t}+xe^t+2$$

(5)强奇性函数(SSN)。

$$f(t,x)=tx-2\sin(t)+\frac{e^t}{x}-4$$

(6)一般非线性项(GSN)。

$$f(t,x)=t+2x^2+4\sin(tx)-\frac{\cos(t)}{x}+1$$

对上述非线性项，我们考虑下列问题：

$$\begin{cases}-x''(t)=f(t,x),t\in(0,1)\\ x(0)=x(1)=0\end{cases}$$

其中 population=20，变异参数为：

$$|x''(t)|\leqslant 10,\omega(x,t)\leqslant 0.1,|x''(0)|\leqslant 15,|x'(0)|\leqslant 10$$

最大误差为0.01。运行结果见下表。

表1 运行结果表

Nonlinear	Max Err	Last Err	Aver Err	Result
SEN	0.01	0.0039	14	OK
SF	0.01	0.0030	3	OK
SP	0.01	0.0099	15	OK
WSN	0.01	0.0083	25	OK
SSN	0.01	0.0078	14	OK
GSN	0.01	0.0020	423	OK

在本文最后，我们给出了系统初始图像，最终图像，错误曲线等。

致谢 本文是在作者访问香港理工大学期间完成的，资助项目有香港理工大学 Research Fellow Matching Fund Scheme 2001 (No. G. YY. 34)，国家自然科学基金，山东省中青年科学家奖励基金项目。

参 考 文 献

1 Ladde G S. Lakshmikantham V. Vatsala A S. Monotone iterative techniques for nonlinear differential equations. Pitman, 1985
2 Heikkila S. Lakshmikantham V. Monotone iterative techniques for discontinuous nonlinear differential equations. Marcel Dekker, Inc., New York, 1994
3 Agarwal R P, Chow Y M. Iterative methods for a fourth order boundary value ptoblems. J. Comput. Appl. Math., 1984 (10): 203～217
4 Buckles B P, Petry F E. Genetic algorithms. Los Alamitos, Calif.: IEEE Computer Society Press, 1992
5 Liu Xiyu. Some existence and nonexistence principles for a class of singular boundary value problems. Nonl. Anal, 1996, 27(10): 1147～1164
6 Guo D, Sun J. Nonlinear Integral Equations. Shandong Science and Technology Press, Ji-Nan, 1987
7 Guo Dajun, Lakshmikantham V. Nonlinear problems in abstract cones. Academic Press, New York, 1988
8 Goldberg D E. Genetic algorithms in search optimization and machine learning. Addison-Wesley, Reading, MA, 1989

(上接第70页)

3. 参考模式阶段中，距离的测量是平方差错误。当存在一个空元时，那个最密集的元包将会被分裂。第一个码本可以通过 N=1而获得；

4. 我们比较测试模式和参考模式的时候，上述的新型的动态时间弯曲方法将会被使用。因为比较的结果直接决定了语音识别的正确性，所以新型的动态时间弯曲是非常重要的；

5. 我们对语音的采样是采用单声道，采样频率是22050赫兹，文件格式采用.wav，去掉了文件头后，将数据经过适当的转化后读入相应的语音库文件；

6. 在识别阶段，我们可以先利用已有的经典动态时间进行衰别，然后设定一个阈值，低于这个阈值的语音被认为是相似语音，再将其通过新型的动态时间弯曲进行识别。

未来的工作 我们已经做的工作是完成了各个阶段的编程实现，我们将要做的工作就是对相似语音进行采样，并且对其分别采用经典的动态时间弯曲和新型的动态时间弯曲，再对其结果进行分析比较，看看结果是不是新型的动态时间弯的性能优于经典的动态时间弯曲。并对这样的结果进行比较分析，得出结论。虽然从理论上来讲后者的性能是优于前者的性能的。

参 考 文 献

1 Kaisheng Y, et al. Residual Noise Compensation For Robust Speech Recognition In Nonstationary Noise, IEEE ICASSP'2000, U. S. A, June 2000
2 邵央，冯哲，李宗葛. HMM 算法框架在银行语音服务中的实现. 计算机工程，2000，26(11)：126～129
3 Rabiner L, Juang B-H. 语音识别基本原理. 清华大学出版社，1999
4 郭军，等. 智能信息处理技术. 北京邮电大学出版社，1999