参考他人报告或代码的申明

统计推断课程，2015 年秋季学期第 12 组，成员 孙天昊 学号 5140309445，张文兵 学号 5140309448，在报告编写过程中，以下方面参考了往届报告，现列表说明：

主要参考项目	说明
代码方面	《统计推断在数模模数转换中的应用》，王博远，2014 年秋季学期，组号 23 在该组报告附录提供的程序代码基础上，进行了修改。

除了以上注明的参考内容，和报告中列出的引用文献注解，本报告其他部分都不包含任何其他个人或集体已经发表或撰写过的作品成果。

统计推断在数模转换系统中的应用

组号：第 12 组　姓名：孙天昊　学号：5140309445　姓名：张文兵　学号：5140309448

摘要：

统计推断是广泛应用于社会科学、自然科学及工程技术研究方面的学科，其模拟方法在实际生产生活中具有重要作用。本课程中，针对大量电子产品样本数据我们将运用统计推断相关知识和 MATLAB 的有关工具，利用遗传算法和模拟退火算法，运用三次样条拟合插值法等确定最佳拟合方式及最优取点方案。利用少量数据的统计特性，合理估算出大量电子产品的特性。

关键词：统计推断，模拟退火算法，遗传算法，MATLAB，多项式拟合法

The application of statistical inference in analog-to-digital conversion system

ABSTRACT：

the statistical inference is widely used in social science, natural science and engineering technology research disciplines, its simulation method plays an important role in the actual production and life. In this course, for a large number of electronic products sample data we will use statistical inference related knowledge and relevant MATLAB tool, using the genetic algorithm and simulated annealing algorithm, using the polynomial fitting method to determine the best fitting method and obtained the optimal solution. Using the statistical characteristics of a small amount of data, a reasonable estimate characteristics of a large number of electronic products.

Key Words： Statistical inference, simulated annealing algorithm, genetic algorithm and MATLAB, polynomial fitting

1. 引言

在工业生产中，我们往往需要对产品的某些特性进行校准和定标。而在定标的过程中，产品的该特性不一定满足简单的线性关系，所以我们需要做的就是确定产品的特性所满足的表达式，并尽可能地使表达式的各系数更加的精确。在这样的校准和定标工序中，我们需要用到多种拟合方法，如三次样条插值拟合法等，确定最符合抽样数据趋势的拟合方式。作为产品特性的拟合曲线。同时，产品数据抽样的取值需要合理算法的支持，如遗

传算法（GA）和模拟退火算法（SA）等。我们需要关注摄取的标准数据点的位置以及数量，来使得定标成本达到最低。通过以上工序，选择最好的方案，来实现对大量工业产品的校准和定标，满足实际的工业生产需求。

2. 课题概述

2.1 模型描述[2]

本课题中的模型是某电子产品内的一个监测模块（图 1），主要讨论的是其中的传感器部件（Sensor）。传感器接收外界的检测物理量 Y，并导出一个电压信号 X。我们需要做的是估计传感器部件的 Y 值与 X 值间一一对应的特性关系，并进行定标。已知，该传感器部件的输入输出特性呈现明显的非线性。

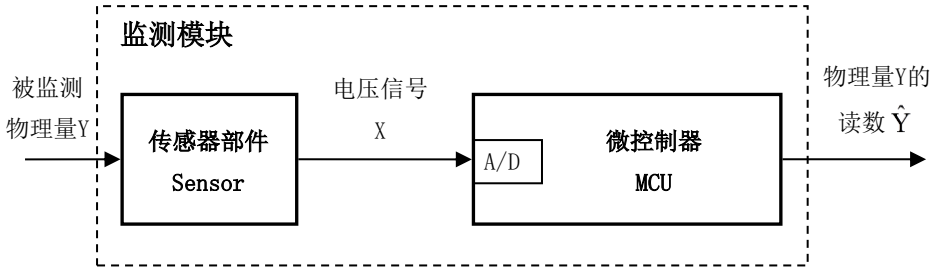


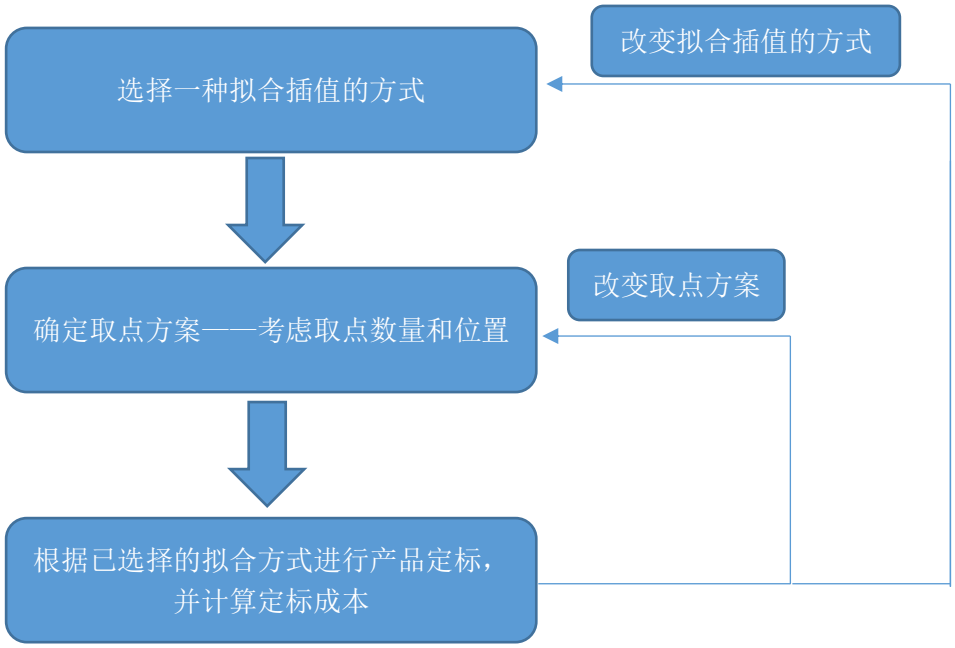
图 1 监测模块组成框图

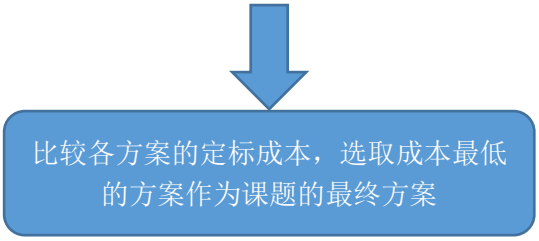
规定：

$$X \in \{x_1, x_2, x_3, ..., x_{50}, x_{51}\} = \{5.0, 5.1, 5.2, ..., 9.9, 10.0\}$$

Y 实测值记为 y_i， $i = 1, 2, 3, ..., 50, 51$

2.2 课题流程图





3. 拟合方法的选择

在定标的过程中，我们主要讨论了以下两种拟合方法：分段三次 Hermite 插值拟合法以及三次样条插值拟合法。从中选取最适合的拟合方式得到最终方案的拟合表达式。在这里，我们选择“遗传算法”来进行两种拟合方法的比较（种群个体数为 200，遗传代数为 100）。

下面是我们分别使用两种拟合方法得到的 4 组最优解：（包括取点情况和最小成本）

表 3-1 分段三次 Hermite 插值拟合法得到的最优解

最优解的取点方案	最小成本
4 16 26 35 48	84.7483
5 17 27 36 49	85.6555
4 17 27 37 49	86.1318
4 15 25 35 48	85.1462

表 3-2 三次样条插值拟合法得到的最优解

最优解的取点方案	最小成本
2 10 21 27 34 44 50	95.4677
2 11 22 30 41 50	95.7603
3 9 20 27 34 44 50	95.3222
2 10 21 30 41 50	95.5603

可以发现，通过分段三次 Hermite 插值拟合法插值得到的最优解对应的最小成本比三次样条插值拟合法的要更低，且程度明显。而且，在时间复杂度上，两种方法的差别并不大。在优先考虑得到最小成本的情况下，本方案选择使用分段三次 Hermite 插值拟合法。

4. 算法实现

针对课题数据，在算法的选择上，我们将采用模拟退火算法以及遗传算法对数据进行分析，以求出最优的模拟方案并进行比较。模拟退火算法和遗传算法均是在统计推断中较为科学的算法工具，其特点如

4.1 遗传算法（GA）

遗传算法是一类借鉴生物进化规律形成的随机化搜索方法。遗传算法的实现可分 为以

下几个步骤：首先随机建立初始状态即个体基因第一代，根据实际问题求解函数来设定适应度函数。其次在接下来的若干代中，依次进行：每个个体适应度函数的求取，根据适应度值的大小进行轮盘概率求取，个体间进行交叉和变异，繁殖出具有适应度更好的下一代。
针对本课题，其具体实现步骤如下：
① 种群初始化:设定 M 为种群个体，G 为遗传代数，生成一个 M*51 的矩阵，矩阵的每一行代表一种取点方式，每一行的每一个元素有两种取值，1 代表选取该点对应的数据，0 代表不选取，由此生成最初始的种群；
② 在种群初始化的基础上，我们在这里提出算法的总体框架，将原始数据保存在 dataY 这个矩阵中，ChosenY 表示最小成本的取点方案，nowMin 表示最小成本。利用循环 for generation=1:G 开始遗传算法，求出每一代的最小成本，并进行代与代之间的比较，将比较得到的最小成本存入 nowMin，并保存对应的取点方案。下一代的生成经过了【根据适应度产生过渡矩阵——adaption.m；过渡矩阵的行与行之间交叉——mutate.m；新一代的每一个个体在一定的变异概率下进行变异——variability.m】。
③ 生成过渡矩阵：我们将父代的成本存入矩阵 m，取 m 中每一个元素的平方的倒数作为每个种群个体的适应度，适应度越大，则成本越小。将适应度最大的个体直接放入过渡矩阵的第一个位置，然后，这里运用了轮盘赌算法来生成后续的过渡矩阵。在使用轮盘法之前，先将矩阵 m 求和，求出每个元素的占比，这样使得每个元素之和为 1，符合概率的要求。我们将 m 的第一个元素与第二个元素相加，替代原来的第二个元素，称之为一个累加过程，这样，当随机数落在 m(i-1)和 m(i)之间时，就说明选中了元素 m(i)。轮盘赌算法的重要语句如下，这里省去了循环的部分：

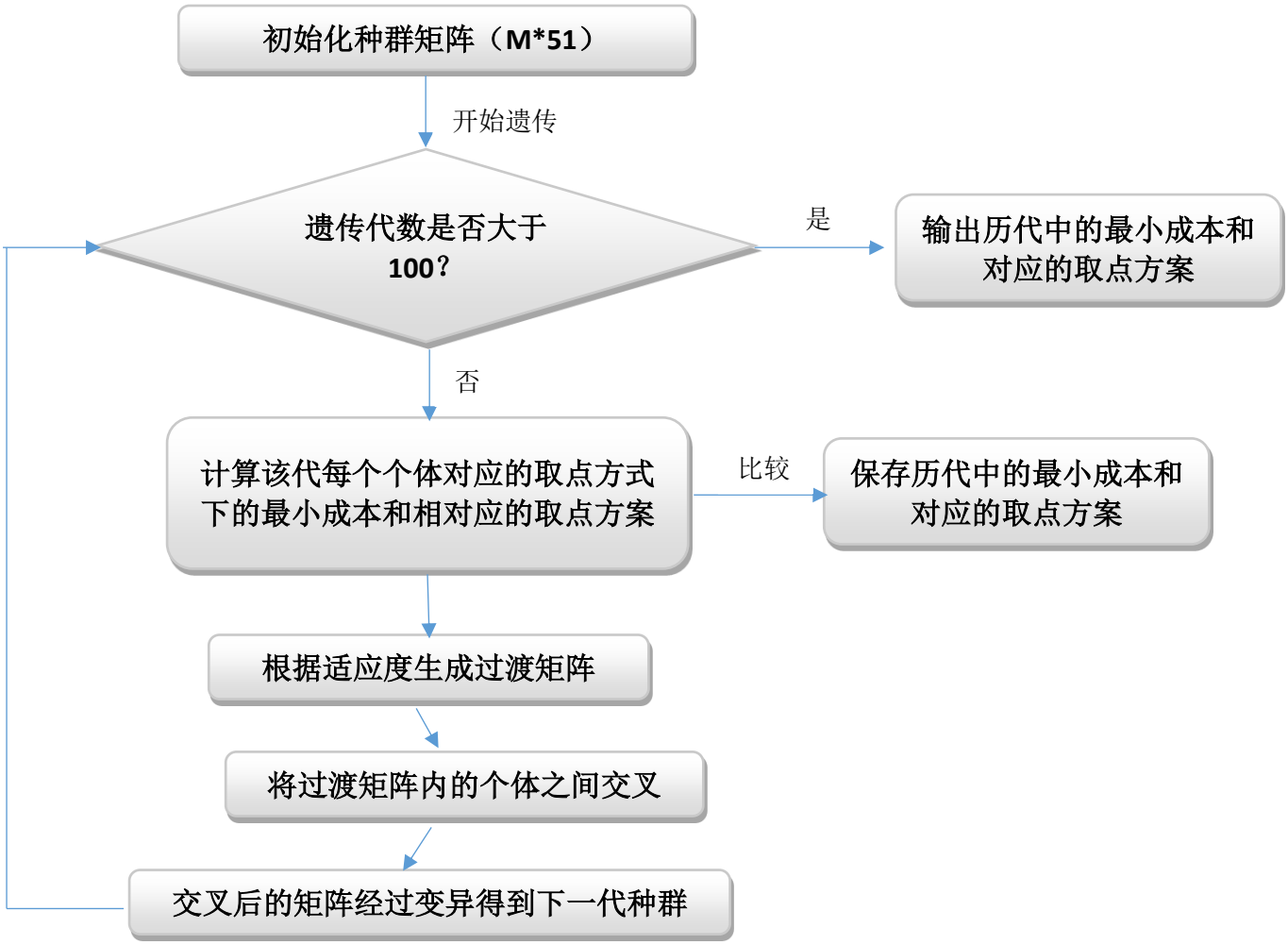
```
c=rand;
if (c < m(j) && c >=m(j-1)) position(i) = j;
```

用 position 矩阵来存储选中的位置，最后将父代中对应位置的个体存储到过渡矩阵中 transition。
④ 交叉过程：这里我们设定的交叉概率为 1，就是，两个个体之间必然会发生交叉。为了保存最优个体，最优个体不参与交叉。交叉的方法是：用 rand 随机产生一个交叉点，将第 i 个个体，和第 M-i+1 个个体之间的片段进行交叉，Positionofmutate 为随机生成的交叉点矩阵，主要代码为：

```
oneofmutate = [mut(i+1,1:positionofmutate(i)),mut(num-
i+1,(positionofmutate(i)+1):51) ];
anotherofmutate = [mut(num-
i+1,1:positionofmutate(i)),mut(i+1,(positionofmutate(i)+1):51) ]
```

如代码所示，假设第 2 个个体和第 200 个个体之间随机生成的交叉点的位置为 26，则取第 2 个个体的 1~26 位置与第 200 个个体的 27~51 位置配对，形成一个新个体存入原来第 2 个个体的位置，而第 2 个个体的 27~51 位置与第 200 个个体的~26 位置配对，形成一个新个体存入原来第 200 个个体的位置，后面的交叉过程同理。但第 101 个个体并没有实现交叉，这个问题是可忽略不计的，对结果的得出并没有什么影响。
⑤ 变异过程：个体变异的实现相对简单，用 rand 生成一个随机数并与变异概率进行比较，这里，我们把变异概率设为 0.01。用两个 for 循环嵌套，对 M×51 矩阵的每一个元素，都生成一个随机数，若随机数小于 0.01，则将这个元素的值改变，改变方式为 0 变为 1 或者 1 变为 0，最后得到的矩阵是经过变异的矩阵，即为子代。

流程图：



4. 2 模拟退火算法（SA）

模拟退火算法来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。根据 Metropolis 准则，粒子在温度 T 时趋于平衡的概率为 e(-ΔE/(kT))，其中 E 为温度 T 时的内能，ΔE 为其改变量，k 为 Boltzmann 常数。用固体退火模拟组合优化问题，将内能 E 模拟为目标函数值 f，温度 T 演化成控制参数 t，即得到解组合优化问题的模拟退火算法：由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解。模拟退火算法其实是在贪心算法的基础上，引入物理退火过程中温度的小概率突变，使其在一定概率内能跳出贪心算法的局限。

① 适应度函数

对所取点的拟合与遗传算法相同

② 成本计算与定标

成本计算函数去课程中教师提供的计算函数，同遗传算法

③ 具体实现过程

首先建立相应的数据矩阵，随机（本实验先建立一个 1x51 的由随机生成的 1 和 0 构成的一阶矩阵，由此作为取点依据，即一阶矩阵数据为 1，取相应点进入计算）取点先得到初始值 J(Y(i))

随后进行变异产生新解，以一定规则进行比较选择，即：

若 J(Y(i+1))>= J(Y(i)) (即移动后得到更优解)，则总是接受该移动
若 J(Y(i+1))< J(Y(i)) (即移动后的解比当前解要差)，则以一定的概率接受移动
根据热力学的原理，在温度为 T 时，出现能量差为 dE 的降温的概率为 P(dE)，表示为：

$$P(dE) = \exp(dE/(kT))$$

其中 k 是一个常数，我们在不断试验中，发现 0.0001 很合适，exp 表示自然指数，且 dE<0（在代码中为 t，也就是两值之差，同时作为判断条件在小于零时执行，故 dE<0）。随着温度 T 的降低（在确定初始温度后，我们采用降温系数 r 对温度降低快慢进行控制，在不断重复过程中，我们取定 r=0.85，T=10000）P(dE)会逐渐降低并随温度降低趋于稳定。
最终得出最佳的取点，并由选定的拟合函数确定取点成本。
为了方便对算法的理解，我们拟出了如下伪码及算法流程图：

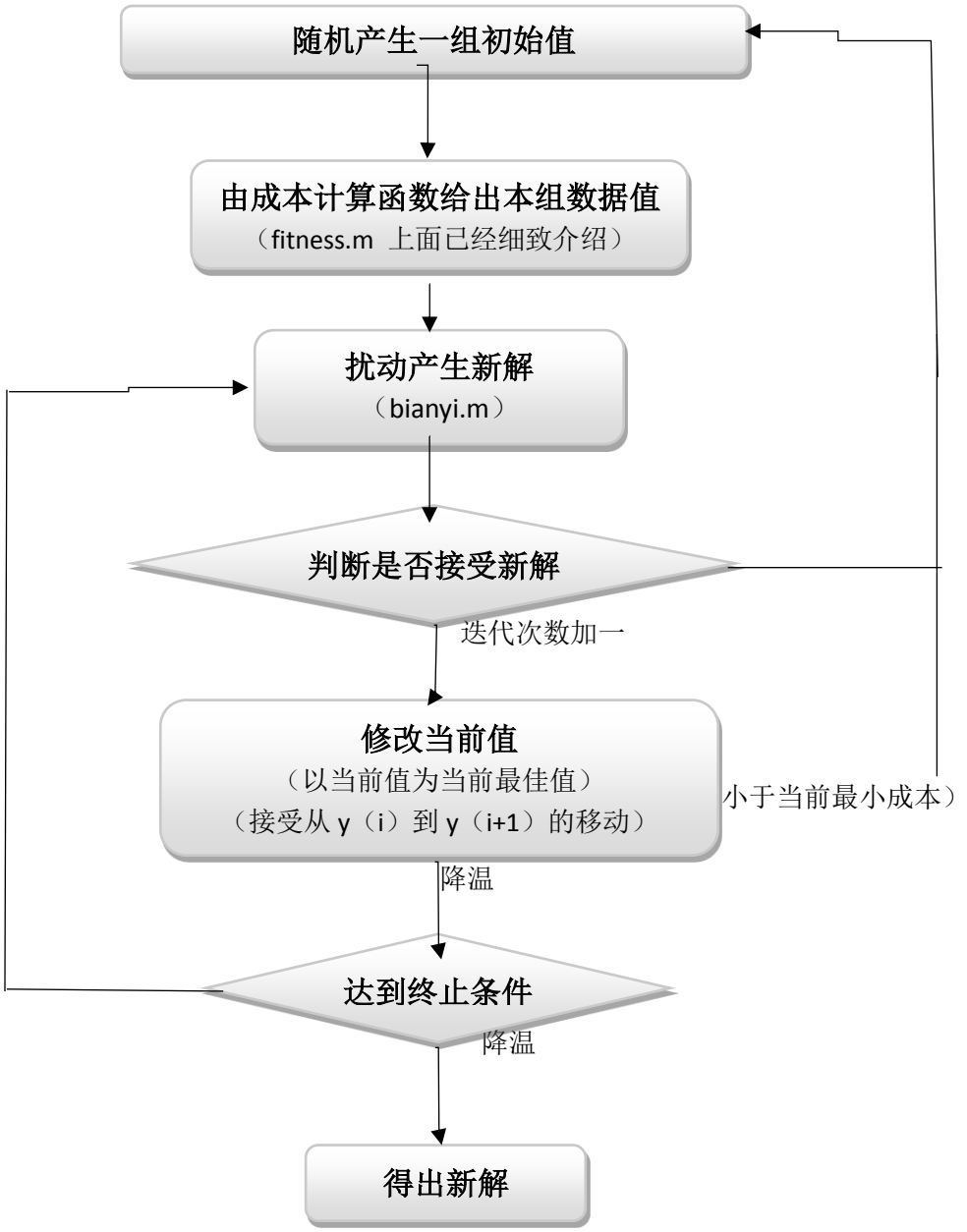
伪码：

* fitness(1,Result1 ,Gene1)：在状态 y 时的评价函数值
*J1：由拟合函数计算的当前取点对应的状态
*J2：由拟合函数计算的变异后新点对应的状态
* r： 用于控制降温的快慢，我们发现取值为 0.85 时恰好能在较小迭代次数前提下达到最优解
* T： 系统的温度，我们将其设为 10000 度
* T_min（0.01） ：温度的下限，若温度 T 达到 T_min，则停止搜索，本次设置为 0.01
*/

```
while( T > 0.01 )
{
    t=J2-J1; （作为变异概率参数，同时作为判断条件）

    if ( t >=0 ) //表达移动后得到更优解，则总是接受移动
J1=J2; //接受从 Y(i)到 Y(i+1)的移动
    else
    {
// 函数 exp( dE/T )的取值范围是(0,1) ， dE/T 越大，则 exp( dE/T )也越大
if ( exp( dE/T ) > rand( 0 , 1 ) )
Y(i+1) = Y(i) ; //接受从 Y(i)到 Y(i+1)的移动
    }
    T = r * T ; //降温退火 ， 0<r<1 。r 越大，降温越慢；r 越小，降温越快
    * 若 r 过大，则搜索到全局最优解的可能会较高，但搜索的过程也就较长。若 r 过小，则搜索的过程会很快，但最终可能会达到一个局部最优值，我们在多次试验之后，将其定为 0.85
        i ++ ;（进行下一次寻找）
}
```

流程图：



另外，虽然穷举法可以将所有可能的数据点选取方案一一列出，且算法比较简单，但过于繁琐，不适用于大数据量的选点，所以在此只做简单说明，本课题不选取该方法进行结果分析。

5. 数据分析与计算结果

在这里，我们选用分段三次 Hermite 插值拟合法，分别运用遗传算法和模拟退火算法进行了 6 次拟合,拟合结果如下：

遗传算法：

种群个体数:200，遗传代数:100，变异率:0.01，保留最优个体，其余个体发生交叉:

表 6-1 遗传算法得到的最优解

最优解的取点方案	最小成本
4 16 26 35 48	84.7483
5 17 27 36 49	85.6555
4 17 27 37 49	86.1318
4 15 25 35 48	85.1462
3 12 21 28 37 49	87.7235
4 15 24 31 39 49	86.6145

模拟退火算法：

初始温度 T：50000，终止温度：0.01，变异率：0.03，接受率：exp(-t/(T*0.001)，退火（降温）因子：0.8

表 6-2 模拟退火算法得到的最优解

最优解的取点方案	最小成本
2 11 21 28 37 49	88.1060
4 14 23 29 37 49	87.1887
3 17 25 35 49	86.5220
4 15 25 35 48	85.1462
3 15 25 35 49	85.6562
3 13 23 29 38 50	87.3242

由以上实验结果可以看到，无论是遗传算法还是模拟退火算法，它们所得到的实验结果都存在着一定的波动性。这个波动性来源于算法中的随机性质。因为随机性的存在和实际应用问题的要求，我们考虑其波动性来分出算法优劣是不正确的。另外，两种算法在运用分段三次 Hermite 插值拟合法时的时间复杂度相差不大，所以时间复杂度也不能作为评判优劣的标准。

从实际问题的角度出发，应选取传感器定标的最小成本，所以我们这份报告选择了遗传算法中的第一组数据作为最终结果，即取点方案为【4 16 26 35 48】，最小成本为 84.7483。即选择了遗传算法作为本课题的求解算法。

6. 实验结果

本报告最终选择了分段三次 Hermite 插值拟合法作为拟合方法，选择遗传算法作为求取最佳取点方案的求解算法。（实际上由于随机性，遗传算法只能求出尽可能优的解）

最佳取点方案：【4 16 26 35 48】

最小成本：84.7483

7. 课题拓展

本课题使用了两种算法来获取最佳的取点方案，在这里，我们选择其中一种算法——遗传算法，来进行进一步的讨论。遗传算法中主要的内设变量为种群个体数 M，遗传代数 G，变异率。

首先，当遗传代数达到 80 的时候，算法得出的最小成本已经趋于稳定，所以只要遗传代数高于 80，则对实验结果不会产生明显影响。

其次是对种群个体数 M 的讨论，这里，我们使用三组数据来说明：

表 7-1 不同种群个体数下的最小成本和取点方案

种群个体数 M	取点方案						最小成本
10	7	11	23	33	42	50	94.7155
200	4	13	22	29	37	49	87.2895
500	4	15	23	30	38	49	86.6187

三组数据都是在遗传 100 代的情况下得出，可以看到，种群个体数越小，最后收敛的成本越大。这是表面上的结论，实际上，由于种群个体数的减小，成本收敛所需的遗传代数是增加的，种群个体数为 10 的情况下，100 代的遗传是不足的。从右侧的图像即可看出，图 7-1 为种群个体数为 10 时的最小成本变化情况。所以，可以得出结论：当种群个体数越小时，越难收敛于最小成本，具体表现为所需的遗传代数增加。

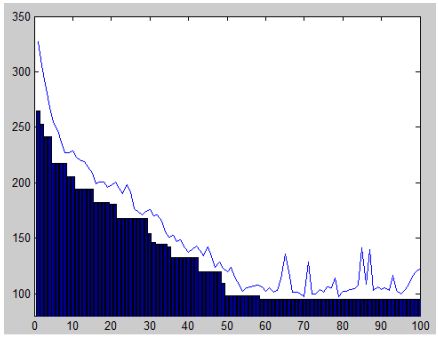


图 7-1 M=10 时最小成本变化情况

再者是对变异率的讨论，我们选择变异率为 1，0.1 和 0.001 三种情况：

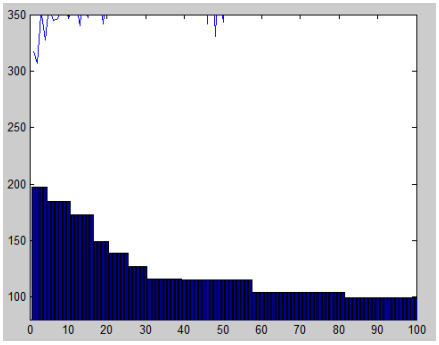


图 7-2 变异率为 1 时最小成本变化情况

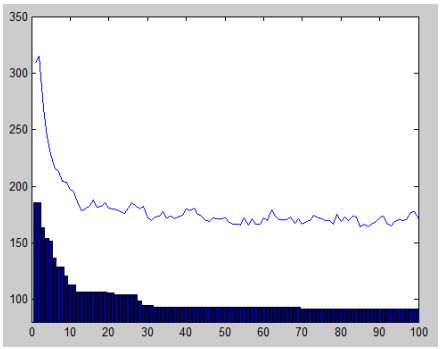


图 7-3 变异率为 0.1 时最小成本变化情况

可以发现，随着变异率的降低，每一代的最小成本的浮动会越来越接近最终的最小成本值。即随着变异率的降低，最小成本浮动降低。同时，若变异率太高，则会影响最终的最小成本的得出，如图 7-2，最小成本甚至会超出 100，与直接得出的实验结论差别甚大。

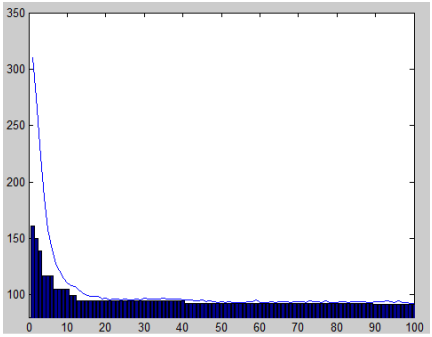


图 7-4 变异率为 0.001 时最小成本变化情况

8. 参考文献

[1] 上海交大电子工程系.“统计推断”课程设计的要求 V2.2 2015-9-22 [EB/OL].ftp://202.120.39.248.

[2] 上海交大电子工程系.“统计推断”课程设计的要求 V2.2 2015-9-22 [EB/OL].ftp://202.120.39.248.

附录（代码清单）：

模拟退火算法：

Main.m 文件

```
tic;
clear all;
clc;
minput=dlmread('20150915dataform.csv');
M=800;
N=51;
samplenum=M/2;
npoint=N;
global x;
x=zeros(1,npoint);
global dataY1;
global dataY2;
average=0;
dataY1=zeros(samplenum,npoint);
Y=zeros(400,51);
M=1;
J1=zeros(1,1);
J2=zeros(1,1);
for i=1:samplenum
    x(1,:)=minput(i*2-1,:);
    dataY1(i,:)=minput(2*i,:);
end
tempMin=12*51+400*51*25;%最小成本临时值，并设定初值
Gene1=round(1*rand(M,51));%随机选定第一代取点集合
T=50000; %初始设定的温度;
generation = 1;
note1=[];
note2=[];
note3=[];
while T>0.1%终止条件，温度减小到最小值
flag2=1;
k=0;
note2=[];
for j=1:90
flag = 0;
Gene2 = variation(Gene1);%变异得到新解
J1 = fitness( 1,J1 ,Gene1);
note2 = [note2,sum(J1)];%求和
```

```
J2 = fitness( 1,J2 ,Gene2);%带入适应度函数求其成本
t=J2-J1;%成本差值，用于判断优劣
if t<0||rand<exp(-t/(T*0.001)) %新解优于原解或以一定的小概率接受新解
    Gene1 = Gene2;
    flag=1;
    if Result2<tempMin
        tempMin=Result2;
        note=dataY2;
    end
end
if flag==1
    k = 0;
else
    k = k+1;
end
if k>40
    flag2=0;
    break;
end
end
average=sum(note2)/90;%求均值
if flag2==0
    break;
end
T%输出当前时间
T=T*0.8;%控制退火（降温)快慢，与迭代次数有关
tempMin%输出当前最优解
note1=[note1,tempMin]; % Record1 记录每一个温度中的最小定标值
note3=[note3,average]; % Record3 记录每一个温度的平均值
note%输出当前取点
generation%输出当前代数
generation=generation+1;
end
bar(note1) %画出最小定标值和平均值的图形
axis([0 60 75 150]);%横坐标由迭代次数有关、纵坐标与最优解的变化有关
hold on;
plot(Record3)
axis([0 60 75 255]);
toc;
```

fitness.m 文件

```
function Result= fitness( ~,Result ,Gene)
```

```
global dataY1;
global x;
global dataY2;
tmp=12*51+400*51*25;
m=1;
[Xtemp,Ytemp]=find(Gene(m,:)~=0);
count=length(Xtemp);
tempX=Ytemp.*0.1+4.9;
ChosenY=dataY1(:,Ytemp);
%由于取点之后你和方式优劣各异，下面写出了三次 i 样条插值、分段三次 hermite 插值法
%并有运行结果知后者更佳
%Y=spline(X1,ChosenY,x); %三次样条插值方法，最终弃用
Y=pchip(tempX,ChosenY,x);  %分段三次 hermite 插值法，最终选择算法
%成本计算函数，取课程中教师提供的成本计算函数
numOfErrorr=abs(calculateY-dataY1);
le0_4=( numOfErrorr<=0.4);
le0_6=( numOfErrorr <=0.6);
le0_8=( numOfErrorr <=0.8);
le1_0=( numOfErrorr <=1);
le2_0=( numOfErrorr <=2);
le3_0=( numOfErrorr <=3);
le5_0=( numOfErrorr <=5);
g5_0=( numOfErrorr >5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-le0_8)+1.5*(le2_0-le1_0)+6*(le3_0-
le2_0)+12*(le5_0-le3_0)+25*g5_0;
si=sum(sij,2);
J(1,m)=sum(si)/400+12* numOfErrorr; %计算误差成本和测定成本
tempM=tempX;
dataY2=Ytemp;
dataY2;
end
```

variation.m 文件

```
function m2 = variation( m2 )
    for i=1:51
        if (rand<0.03)%  取较小的变异概率符合退火的物理过程
          m2(1,i) = 1-m2(1,i);
        end
    end
end
```

遗传算法：

Main.m 文件

```
tic;
clear all;clc;
outfile = fopen('output.txt','wt');
minput=dlmread('20150915dataform.csv');
[M0,N]=size(minput); %读取样本数量和取样点数量
samplenum=M0/2;npoint=N;
global dataY;%原始样本数据
global ChosenY;
global nowMin; %截至到第 i 代时最小的成本
global average;%每一代的最小成本
dataY=zeros(samplenum,npoint);
fitY=zeros(samplenum,npoint);
M=200; %种群数量
G=100; %遗传代数
Result=zeros(1,M);
for i=1:samplenum
    dataY(i,:)=minput(2*i,:);
end
Min=[];
tempaverage=[];
Gene=round(1*rand(M,51)); %群体基因初始化
for generation=1:G %开始遗传算法
    Result = fitness(M,Result,Gene); %计算成本
    transit=adaption(M,Result,Gene); %进行适应度选择，生成过渡矩阵
    son = mutate(transit,M); %交叉
    variason = variability(son,M); %变异
    Gene=variason;
    generation
    ChosenY
    nowMin
    Min=[Min,nowMin];
    tempaverage=[tempaverage,average];
    fprintf(outfile,'generation:%d \n ',generation);%输出结果到输入文件
    fprintf(outfile,'min_cost:%.4f \n',nowMin);
    average
end
bar(Min)%作图
axis([0 100 80 350]);
hold on;
plot(tempaverage)
```

```
axis([0 100 80 350]);
toc;
```

mutate.m 文件

```
function mut = mutate(mut,num)
% 交叉函数
numofmutate =num/2; %交叉次数
positionofmutate = randi(51,numofmutate,1); %随机生成交叉的位置
for i = 1:1:numofmutate %交叉 numofmutate 次
    if (positionofmutate(i)==51)
        continue
    end
    oneofmutate = [mut(i+1,1:positionofmutate(i)),mut(num-i+1,(positionofmutate(i)+1):51) ]; %
交叉过程
    anotherofmutate = [mut(num-
i+1,1:positionofmutate(i)),mut(i+1,(positionofmutate(i)+1):51) ];
    if (i~=1)
        mut(i,:) = oneofmutate; %保留最优个体在位置 1
    end
    mut(num-i+1,:) = anotherofmutate; %交叉后的个体
end
out = mut; %返回下一代
end
```

Variability.m 文件

```
function final = variability( final ,num)
for i=1:51
    for j=2:num %最优个体不变异
        if (rand<0.01) %变异概率为 0.01
            final(j,i) = 1-final(j,i);
        end
    end
end
out = final;%返回新的种群
end
```

adaption.m 文件

```
function transition = adaption(M,m,parent)
%适应度函数，用于生成过渡矩阵
m=m.^2;
```

```
m=1./m;%平方后取倒数
sum1=sum(m);
m = 1/sum1 *m;%使得 m 求和为 1，作为概率
position = zeros(1,M)+1;%生成一个 1*M 的矩阵分别对应 M 个种群个体
transition = zeros(M,51);%初始化过渡矩阵
leastResult=m(1);%先设定过渡矩阵的第一个元素为最小成本
minpos=1;%最小适应度对应的位置
for i=2:1:M
    if (m(i)>leastResult) %找到最大的 m(i)，即为最小成本
        minpos = i;%记录最小成本在 M 个个体中的位置
        leastResult=m(i);
    end
    m(i)= m(i-1)+m(i);
end
for i=2:1:M %将第一个位置留给最优个体（轮盘赌算法）
    for j=1:1:M %对应于过渡矩阵的 M 个种群个体，需要 M 个骰子
        c=rand; %产生随机数
        if (c < m(j) && j==1) position(i) = j ; %总的来说，哪个区域段的概率越大，越容易被选中，其对应的成本越低，第一个点由于 m-1 不存在，单独处理
            continue;
        end
        if (c < m(j) && c >=m(j-1)) position(i) = j ; %骰子落在第 j 个区域，说明第 j 个种群个体被选中，记录位置 j，并跳出循环
            continue;
        end
    end
end
transition(1,:) = parent(minpos,:); %将最优个体放在过渡矩阵的第一个位置
for i=2:1:M
    transition(i,:) = parent(position(i),:); %生成过渡矩阵的所有元素，越靠前的成本越低
end
out=transition; %返回过渡矩阵
end
```

fitness.m 文件

```
function Result= fitness(M,Result,Gene)
xi=[5.0:0.1:10.0];%定义插值点
global dataY;
global nowMin;
nowMin=12*51+400*51*25;
global ChosenY; %最小成本对应的取点方案
global average;
average=0;
```

```
for m=1:M
    [tmpx,tmpy]=find(Gene(m,:)~=0);
    numofSelectedData=length(tmpx);%统计测试的样本点数量
    xrange=tmpy.*0.1+4.9;%定义插值范围
    yrange=dataY(:,tmpy);
    fittingY=pchip(xrange,yrange,xi); %分段三次 hermite 插值法
    errorofData=abs(fittingY-dataY); %取误差大小
    le0_4=(errorofData<=0.4);%误差成本列出
    le0_6=(errorofData<=0.6);
    le0_8=(errorofData<=0.8);
    le1_0=(errorofData<=1);
    le2_0=(errorofData<=2);
    le3_0=(errorofData<=3);
    le5_0=(errorofData<=5);
    g5_0=(errorofData>5);
    sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-le0_8)+1.5*(le2_0-le1_0)+6*(le3_0-
le2_0)+12*(le5_0-le3_0)+25*g5_0;%单点误差成本
    si=sum(sij,2)+12*ones(400,1)*numofSelectedData;
    Result(1,m)=sum(si)/400; %计算平均成本
    if Result(1,m)<nowMin
        ChosenY=tmpy;
        nowMin=Result(1,m);%保留最低平均成本
    end
end
average=sum(Result)/M;
out=Result;
end
```

%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%

test_ur_answer.m

```
my_answer=[4,16,26,35,48];%把你的选点组合填写在此
my_answer_n=size(my_answer,2);

% 标准样本原始数据读入
minput=dlmread('20150915dataform.csv');
[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample,npoint);
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
```

```
for i=1:nsample
    x(i,:)=minput(2*i-1,:);
    y0(i,:)=minput(2*i,:);
end
my_answer_gene=zeros(1,npoint);
my_answer_gene(my_answer)=1;


% 定标计算
index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample
    % 请把你的定标计算方法写入函数 mycurvefitting
    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

% 成本计算
Q=12;
errabs=abs(y0-y1);

le0_4=(errabs<=0.4);
le0_6=(errabs<=0.6);
le0_8=(errabs<=0.8);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-le0_8)+1.5*(le2_0-le1_0)+6*(le3_0-
le2_0)+12*(le5_0-le3_0)+25*g5_0;
si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
cost=sum(si)/nsample;

% 显示结果
fprintf('\n 经计算，你的答案对应的总体成本为%5.4f\n',cost);
```

mycurvefitting.m

```
function y1 = mycurvefitting( x_premea,y0_premea )

x=[5.0:0.1:10.0];
```

```
% 将你的定标计算方法写成指令代码，以下样式仅供参考
y1=interp1(x_premea,y0_premea,x,'pchip');


end
```