

# 一个测量装置在大规模制造中的标定问题

第一组 whh allesgutewh@gmail.com

## 摘要

本课题意在求解一个温度测量装置在大规模制造中标定成本最小化方案。考虑到标定的测量成本和误差成本，合理选择测量点的个数和测量标定的温度，并在测量点的基础上利用插值拟合完成标定。

本文选用了遗传算法对测量点的选取做优化，利用插值法完成标定，通过不断改善种群进化细节、插值标定方式，得到该温度测量装置标定成本最小化方案。

**关键词：**测量装置标定，三次样条插值，Hermite 插值法，遗传算法

## Calibration of a Measuring Device in Large-scale Manufacturing

### ABSTRACT

The purpose of this subject is to solve the calibration cost minimization scheme of a temperature measuring device in large-scale manufacturing. Considering the measurement cost and error cost of calibration, the number of measurement points and the temperature of the measurement calibration are reasonably selected, and interpolation is used to complete the calibration based on the measurement points.

In this paper, the genetic algorithm is used to optimize the selection of measurement points, and the interpolation method is used to complete the calibration. By continuously improving the details of population evolution and interpolation calibration methods, the calibration cost minimization scheme of the temperature measurement device is obtained.

**Keywords:** calibration of measuring device, spline interpolation, Hermite interpolation, genetic algorithm

## 1 问题重述

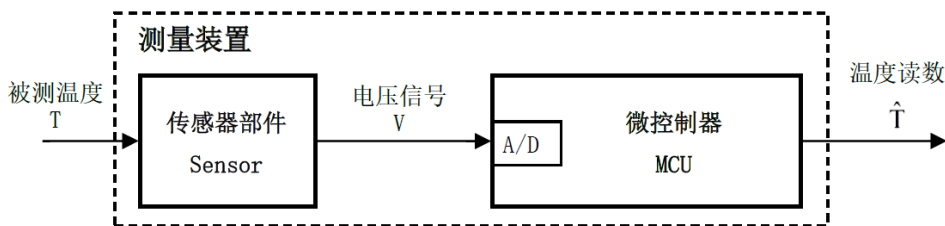
### 1.1 问题背景

很多电子产品中带有测量功能，用于监测某种物理量，传感器所收集到的信号一般不能直接反映被测量的物理量，这就涉及到标定问题。标定就是得到传感器收集到的信号与被测物理量的值的对应关系。即使是用于同一物理量测量的传感器也存在个体差异，不能用同一标准标定。

在大规模生产的过程中，不仅要考虑准确度，还要考虑标定成本。实际生产中往往不需要对每个点都进行测量，而是选择部分点准确测量，然后采取拟合插值的方式确定其他点的对应关系。随着准确测量点的个数增加，标定的可靠性提升、误差成本降低，准确测量产生的成本上升；准确测量点的分布也会影响误差。要实现成本最小化就要合理选择测量点的个数和分布。

### 1.2 问题的提出

本课题研究的是温度测量装置，可测范围是  $-20^{\circ}\text{C}$  至  $70^{\circ}\text{C}$ 。所用核心传感器的输入-输出特性有明显非线性，且个体差异性比较明显。

图 1: 温度测量装置原理框图<sup>[1]</sup>

本课题的目标是针对图1所示温度测量装置，利用已有的数据得到适合大规模高效率批量制造的标定方式。

## 2 模型假设

1. 测量任意一点的温度成本相同且为固定值。
2. 成本仅考虑测量成本和误差成本。

## 3 符号说明

表1列出了本文需要的符号。

表 1: 符号说明

符号	符号描述
$V$	传感器的输出电压
$T$	传感器的输入，即被测温度
$\hat{T}$	测量装置的输出 (读数)
$T_{i,j}$	第 $i$ 个个体第 $j$ 个点的测量温度 (来自标准样本数据库)
$\hat{T}_{i,j}$	第 $i$ 个个体第 $j$ 个点的标定温度
$Q$	单点标定成本，指定为 50
$s_{i,j}$	第 $i$ 个个体第 $j$ 个标定点的误差成本
$S_i$	第 $i$ 个个体的误差成本
$n_i$	第 $i$ 个个体的测量点数
$C_i$	第 $i$ 个个体的标定总成本
$NUM$	种群规模，即一代种群中的个体数
$GEN$	最大繁衍代数
$C_r$	交叉率
$V_r$	变异率 (每个个体发生单点变异的概率)

## 4 理论基础

### 4.1 插值拟合

拟合是用一个函数刻画离散点，拟合得到的函数不一定经过所有离散点；插值是在离散数据的基础上补插连续函数，使得这条连续曲线通过全部给定的离散数据点<sup>[2]</sup>。插值、拟合的方式有很多，图2展示了

其中三种拟合插值方法：二次多项式拟合 (polynomial)、Hermite 插值 (pchip)、三次样条插值 (spline) 得到的曲线对比，可知针对此问题二次多项式拟合较其他两种效果差。

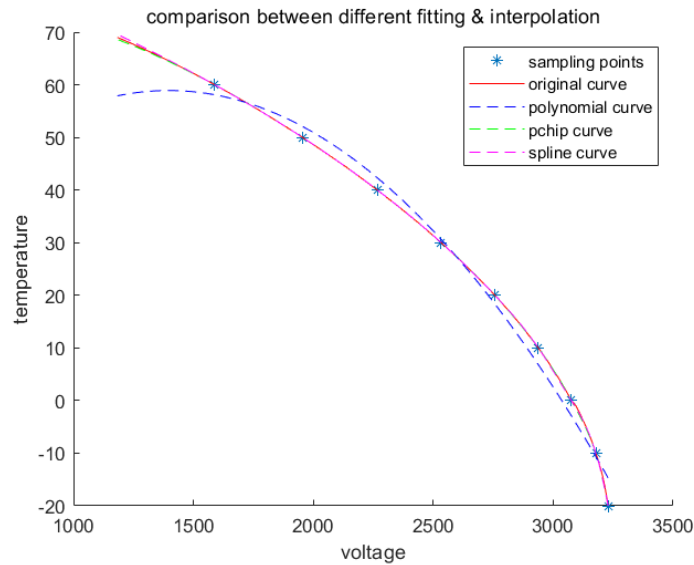


图 2: 不同拟合、插值方法得到的曲线对比

**Hermite 插值** 为了使插值函数能更好地逼近被插值函数，不但要求二者在节点上函数值相等，而且还要求插值函数在某些节点或全部节点上与被插值函数的导数值也相等，甚至要求高阶导数也相等<sup>[3]</sup>。

**三次样条插值** 三次样条插值用节点处二阶导数表示三次样条插值函数，用追赶法求解相关方程组。通过 Matlab 编制三次样条函数的通用程序，可直接显示各区间段三次样条函数体表达式，计算出已给点插值并显示各区间分段曲线图<sup>[4]</sup>。

本文选取部分温度点及测量值作为初始离散数据，通过三次样条插值获得其他电压值对应的温度，从而得到一个样本 90 组电压与温度的关系完成标定。

在拓展探究部分将探究 Hermite 插值法在本问题中的应用以及使用两次插值进行标定的方法。

4.2 遗传算法

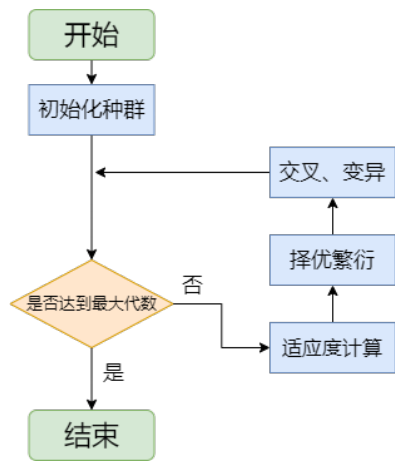


图 3: 遗传算法基本流程图

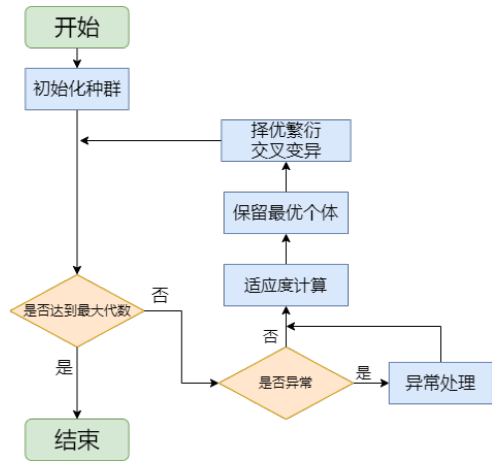


图 4: 本案例遗传算法流程图

遗传算法 (Genetic Algorithm) 是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法<sup>[5]</sup>。以本案例为例，在最少选择两个标定点的

基础上标定点的选取方式共有  $(2^{90} - 91)$  种, 即使用枚举法要计算  $(2^{90} - 91)$  种情况; 而使用遗传算法种群规模为 64、最大繁衍代数 64 的条件下只需计算 4096 种情况。前者的数量级为  $10^{27}$ , 后者数量级仅为  $10^3$ , 由此可得遗传算法在本问题最优解的搜索中优势明显。

如图3所示遗传算法基本流程包括种群初始化、适应度计算、择优繁衍、交叉、变异。图4所示为本文中所使用的遗传算法流程图, 其中异常检测是为确保三次样条插值可以正常进行, 针对具体细节可以对每个步骤的算法进行调整以达到更高的搜索效率。

## 5 解决方案

### 5.1 数据

数据来自课程组提供的数据集<sup>[1]</sup>, 其中用于训练的数据文件有 1000 行, 每行 90 个数据对应  $-20\text{ }^{\circ}\text{C}$  至  $70\text{ }^{\circ}\text{C}$  的温度 ( $T$ ) 和电压值 ( $V$ ); 奇数行是温度数据 (单位:  $^{\circ}\text{C}$ ), 偶数行是电压数据 (单位:  $\text{mV}$ ), 每两行代表一个传感器的数据, 共有 500 个传感器的数据。这些数据将作为真实测量值的来源及评价成本的依据。

### 5.2 模型细节

如图5所示, 在本问题中每个个体有 90 个基因代表 90 个温度点, 每个基因有 0、1 两种状态, 0 代表不使用该点的真实测量值、1 代表使用该点的真实值。

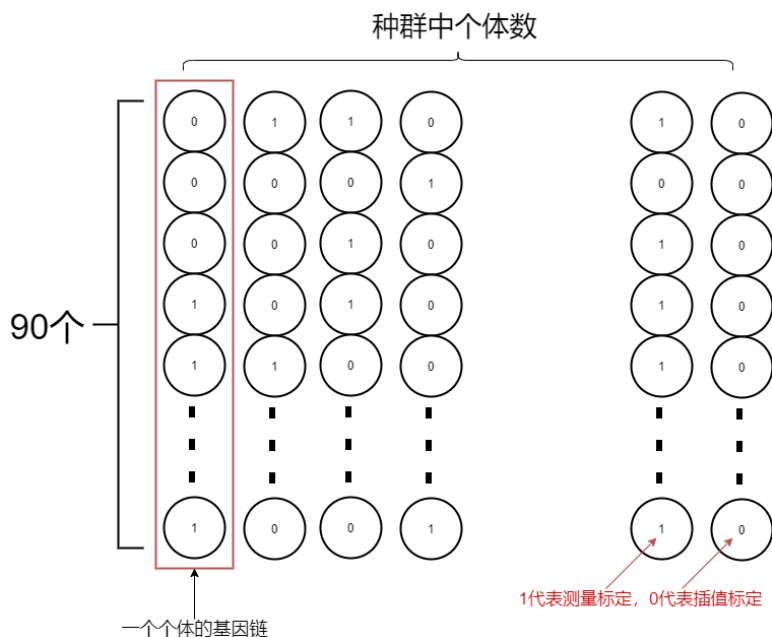


图 5: 一代种群的组成

首先, 初始化种群的规模 ( $NUM$ )、最大繁衍代数 ( $GEN$ )、交叉率 ( $C_r$ )、变异率 ( $V_r$ ), 并用随机数为第一代种群赋初值。此后种群将进入繁衍的循环, 每一代种群经过适应度计算、择优繁衍、交叉、变异得到下一代。在这个过程中记录每一代的最优个体的成本, 最终输出最后一代的最优个体。

在本问题求解过程中, 种群规模  $NUM = 64$ , 繁衍代数  $GEN = 64$ , 交叉率  $C_r = 0.7$ , 变异率  $V_r = 0.3$ 。

下面详细介绍一代种群的繁衍过程。

#### 成本计算

使用每个个体基因链中为 1 的点作为测量点其他点进行三次样条插值完成标定。一个个体的标定成

本包括测定成本和误差成本。单点测定成本  $Q$  指定为 50，单点误差成本按式1计算。

$$s_{i,j} = \begin{cases} 0 & |\hat{T}_{i,j} - T_{i,j}| \leq 0.5 \\ 1 & 0.5 < |\hat{T}_{i,j} - T_{i,j}| \leq 1.0 \\ 5 & 1.0 < |\hat{T}_{i,j} - T_{i,j}| \leq 1.5 \\ 10 & 1.5 < |\hat{T}_{i,j} - T_{i,j}| \leq 2.0 \\ 10000 & |\hat{T}_{i,j} - T_{i,j}| > 2.0 \end{cases} \quad (1)$$

得到第  $i$  个个体的标定总成本为  $c_i = S_i + Q \cdot n_i$ ，其中  $S_i = \sum_{j=1}^{90} s_{i,j}$  为总误差成本 (500 个传感器分别标定获得的误差成本平均值)、 $n_i$  为测量点个数。

### 择优繁衍

在本问题中，成本越低的个体适应度越高。采取了成本取相反数操作，此时适应度越高成本越低。为了保证适应度的区分度，采取所有个体成本减最小值然后除以极差的方式，将种群中全部个体的适应度映射到区间  $[0, 1]$ ，此后根据适应度占比决定个体繁衍后代的比例。

在实验过程中发现，由于交叉和变异，最后一代的最优个体有一定的概率不是各代种群的最优个体，因此采取了优化措施将每一代种群的最优个体作为下一代的最后一个个体，各代种群最后一个个体不参与后续的交叉和变异。

### 交叉

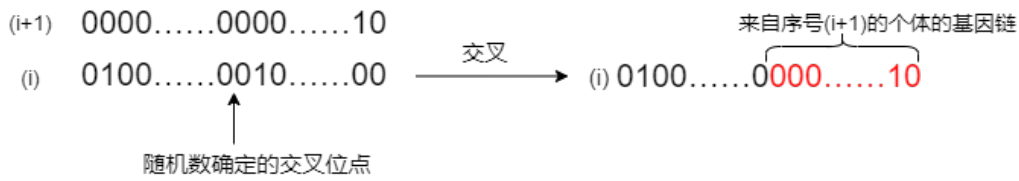


图 6: 第  $i$  个个体交叉示意图

根据交叉率  $C_r$  决定每个个体是否与相邻个体发生交叉，采用单点交叉。如图6所示，通过随机数确定交叉位点，在此点之后的基因序列替换为相邻个体对应的基因链。

### 变异

根据变异率  $V_r$  决定一个个体是否发生变异，采用单点变异，通过随机数决定变异个体变异的位点。

由于保留了上一代最优个体，可以适当增大变异率，因此本问题求解的过程中将变异率设置为一个相对于自然种群变异率较高的值 0.3。

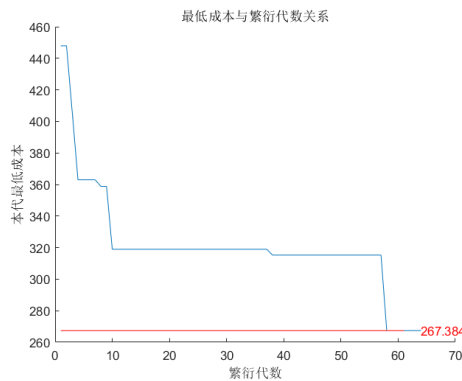


图 7: 最低成本与繁衍代数关系 1

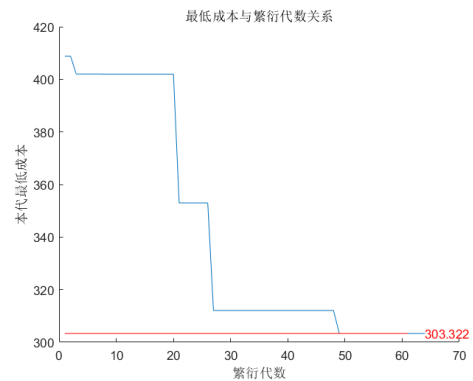


图 8: 最低成本与繁衍代数关系 2

在上述算法的基础上求解，运行两次的结果如图7、8所示，两次得到的最优解成本相差甚远，其中图7得到的最优解选取的测量点个数为 5，8得到的最优解选取的测量点个数为 6，显然图五所示的 5 个测量点更接近最优结果，这为模型的优化提供了思路。

5.3 模型优化

实验过程中出现的最优解成本小于 300，因此当  $n_i \geq 6$  时增多测量点即基因由 0 变异为 1 对于寻找最优解的帮助不大。对变异方式进行优化，令  $n_i \geq 6$  的个体只能产生由 1 到 0 的变异，其他个体仍然采用随机单点变异。

优化后的程序多次运行都可以得到 270 左右的最低成本，图9为其中一次求解的结果，选择的测量点温度值表2所示。

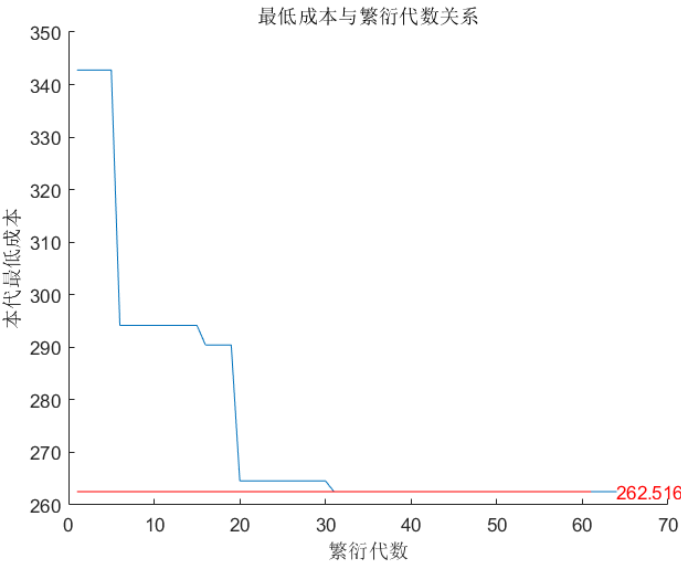


图 9: 优化后的最低成本与繁衍代数关系

表 2: 最优解测量点

序号	1	2	3	4	5
测量点温度 (°C)	-16	-2	7	52	67

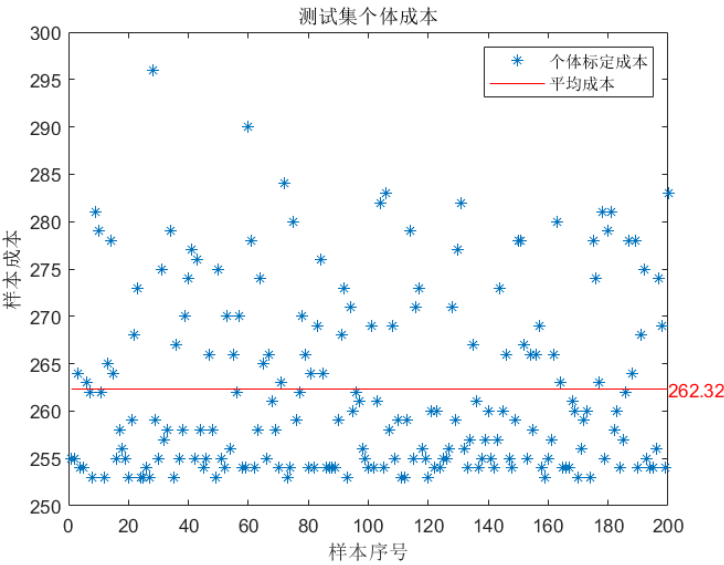


图 10: 测试数据成本

将这组解应用于测试数据集，得到如图10所示的成本，其中平均成本为 262.32。与训练数据相差不大，可以认为结果可靠。

6 拓展研究

6.1 不同插值函数的探究

使用在4.1中提到的 Hermit 插值法代替三次样条插值法进行计算，在求解过程中出现了低于 250 的最优成本，此时的最优方案测量点个数不超过 4。在此基础上对变异方式进行相应优化，令  $n_i \geq 4$  的个体只能产生由 1 到 0 的变异，其他个体仍然采用随机单点变异。此时得到的最优解平均成本在 230 左右，表3列举了几次求解的结果，由表中数据可知，训练集与测试集平均成本相当，认为结果可靠。

表 3: Hermite 插值法标定方案

序号	测量点温度 (°C)				训练集平均成本	测试集平均成本
1	-16	-5	48	67	227.6	227.4
2	-16	-6	45	68	232.6	232.4
3	-17	-6	48	68	233.2	233.1
4	-17	-4	53	62	225.3	225.0

取其中成本最低的一组温度值 -17 °C、-4 °C、53 °C、62 °C，作为此方案条件下的最优解，其对应测试集成本如图11。

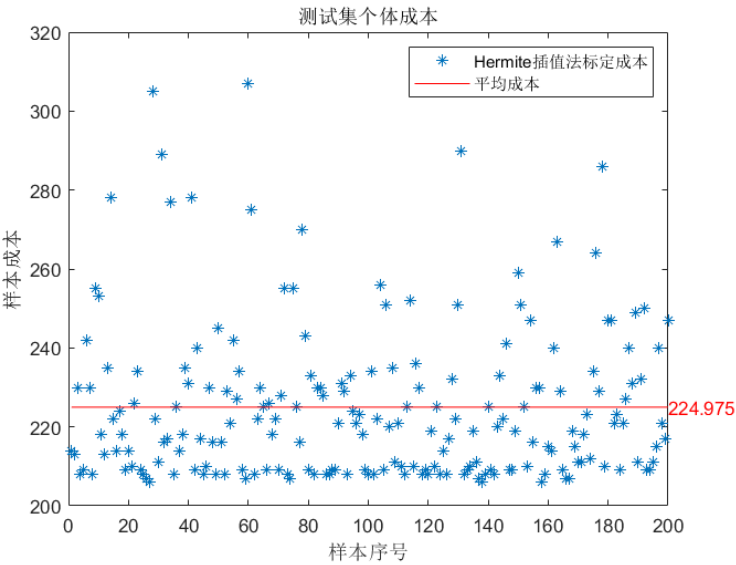


图 11: Hermite 插值法最优解测试数据成本

6.2 不同标定方法的探究

受到段以恒<sup>[6]</sup>研究的启发，使用了一种新的标定方式，具体流程如下：

- 1. 根据已知点以温度为已知量进行第一次插值，得到与各温度值对应的电压值；
- 2. 根据第一次标定得到的温度、电压值对应值以电压为已知量进行第二次插值，得到与电压对应的温度值；

3. 根据第二次标定得到的温度值与实际温度值计算误差成本。

表 4: 二次标定方案

序号	测量点温度 (°C)			训练集平均成本
1	-17	31	62	156.1
2	-15	0	44	167.3
3	-15	35	59	159.0
4	-12	39	58	162.6

表4列举了部分运行结果，根据表中数据可知最优解的测量点个数为 3，本问题中所有可能取值有  $C_{90}^3 = 117480$  种，可采用枚举法获得最优解。不妨假设第一个温度点的取值在  $20^{\circ}\text{C}$  之前，各温度点间隔不小于  $5^{\circ}\text{C}$ ，满足此条件的方案共 76480 种，枚举得到最优解的测量点在  $-15^{\circ}\text{C}$ 、 $21^{\circ}\text{C}$ 、 $64^{\circ}\text{C}$ ，训练集、测试集平均成本分别为 152.7、152.7，其中测试集数据成本如图12所示。

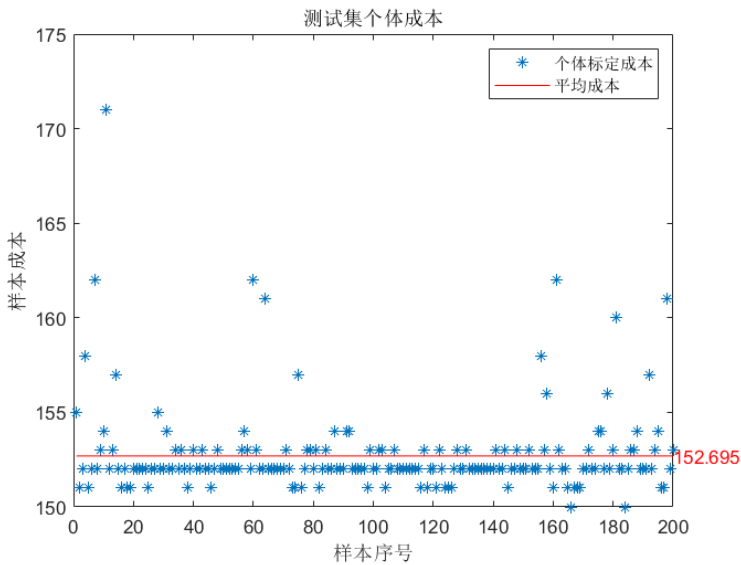


图 12: 二次插值法最优解测试数据成本

7 结论

- 1. 在测量装置在大规模制造中的标定问题中，根据少量点利用插值完成标定可以有效减小成本。
- 2. 遗传算法在选择合适的标定点、实现成本最小化的过程中非常有效。通过遗传算法得到的最优解在测试数据中的平均成本与训练数据的平均成本非常接近。
- 3. 本问题的最优方案为选择三个点进行测量，先后进行两次三次样条法插值完成标定，三个点的温度分别为  $-15^{\circ}\text{C}$ 、 $21^{\circ}\text{C}$ 、 $64^{\circ}\text{C}$ 。



## 参考文献

- [1] 上海交通大学电子工程系, 一个测量装置在大规模制造中的标定问题 [DB/OL], ftp://202.120.39.248, 2020.
- [2] 《数学手册》编写组, 数学手册 [M]. 高等教育出版社, 1979.
- [3] 王艳, “关于埃尔米特插值的教学探讨,” 重庆与世界 (学术版), vol. 32, no. 03, pp. 8–10+13, 2015.
- [4] . 许小勇, “三次样条插值函数的构造与 matlab 实现,” 兵工自动化, no. 11, pp. 81–83.
- [5] “遗传算法,” Baidu. [Online]. Available: <https://baike.baidu.com/item/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95>
- [6] 段以恒, 一个测量装置在大规模制造中的标定问题 [OL], ftp://202.120.39.248, 2020.

## 附录

## A 求解代码

```
1 %读取数据
2 minput=dlmread('dataform_train.csv');
3 %构造温度、电压矩阵
4 global all_temp all_volt;
5 all_temp=minput(1:2:1000,:);
6 all_volt=minput(2:2:1000,:);
7 [M,N]=size(all_volt);
8 %%%遗传算法基本参数
9 NUM=64;          %种群规模
10 GEN=64;          %代数
11 cross_rate=0.7;  %交叉率
12 change_rate=0.3; %变异率
13
14 %记录每一代的成本
15 cos_ave=zeros(NUM,GEN);
16 %记录每一代的最优个体
17 max_p=zeros(N,GEN);
18 %%%初始化种群
19 groups=zeros(N,NUM,GEN);%保留每一代的种群，每个个体N个基因、每一代种群有NUM个个体，一共GEN
    代
20 %初始化第一代种群
21 groups(:,:,1)=rand(N,NUM,1);
22 threshold=0.9;
23 groups=(groups>threshold);
24 for i = 1:GEN
25     choice=groups(:,:,i);
26     %异常处理
27     error=sum(choice);
28     error=find(error<3);
29     if error
30         choice=t(choice,error);
31     end
32     %计算成本
33     cos_ave(:,i)=f(choice);
34     %保留最优个体
35     ppp=choice(:,cos_ave(:,i)==min(cos_ave(:,i)));
36     max_p(:,i)=ppp(:,1);
37     %根据这一代选择下一代
38     choice=selection(choice,cos_ave(:,i));
39     choice(:,NUM)=max_p(:,i);
```

```
40     %cross
41     choice=ex(choice,cross_rate);
42     %variation
43     groups(:,:,i+1)=var(choice,change_rate);
44 end
45 %输出
46 find(max_p(:,GEN))-20
47 y=min(cos_ave);
48 hold on
49 plot(1:GEN,y);
50 xlabel('繁衍代数');ylabel('本代最低成本');
51 title('最低成本与繁衍代数关系');
52 plot(1:4:GEN,y(GEN)*ones(GEN/4),'r');
53 text(GEN,y(GEN),[num2str(y(GEN))],'color','r');
```

## B 计算成本函数

```
1 function [fitness]=f(choice)
2 %输入这一带的choice
3 %输出每个个体对应的fitness
4 global all_volt all_temp
5 s=[];
6 fitness=50*sum(choice);
7 [~,NUM]=size(choice);
8 [M,N]=size(all_volt);
9 for i =1:NUM
10     p=zeros(M,N);
11     for j =1:M
12         p(j,:)=spline(all_volt(j,choice(:,i)),all_temp(j,choice(:,i)),all_volt(j,:));
13     end
14     p=abs(p-all_temp);
15     s=[s,(length(find(p>0.5&p<=1))+length(find(p>1&p<=1.5))*5+length(find(p>1.5&p<=2))
16         *10+length(find(p>2))*10000)/500];
17 end
18 fitness=fitness+s;
19 end
```

## C 择优繁衍函数

```
1 function [newchoice] = selection(choice,cos)
2 x= length(cos);
3 newchoice=choice;
4 cos=-cos;
```

```
5 a=max(cos);
6 b=min(cos);
7 value=(cos-b)/(a-b);
8 totalvalue = sum(value);
9 %计算各个个体的价值占比
10 value_rate = value/totalvalue;
11 value_rate = cumsum(value_rate);%概率求和(得到0-1之间的累积量)
12 cur = 1;%目前的个体
13 i = 1;
14 while i<=x&&cur<=x
15     %如果当前面积小于已处理的个体累计占比, 新种群里增加一个当前个体
16     if(i/x)<value_rate(cur)
17         newchoice(:,i)=choice(:,cur);
18         i = i+1;
19     else%比例持平时, 处理下一个个体
20         cur=cur+1;
21     end
22 end
23 %打乱顺序
24 r=randperm(x);
25 newchoice=newchoice(:,r);
26 end
```

## D 交叉函数

```
1 function [newchoice] = ex(choice,exchange)
2 [x,y] = size(choice);
3 %初始化新的选择组合
4 newchoice = choice;
5 %两个一组交换
6 for i = 1:y-1%最后一个不参与交换
7     if(rand<exchange)%如果交换
8         expoint = round(rand*x);%随机确定交换的位置
9         newchoice(:,i) = [choice(1:expoint,i);choice(expoint+1:x,i+1)];
10    end
11 end
12 end
```

## E 变异函数

```
1 function [newchoice] = var(choice,vari)
2 [x,y] = size(choice);
3 %初始化新的组合
```

```
4 newchoice = choice;
5 for i = 1:y-1
6     if(rand<vari)%发生变异
7         a=sum(choice(:,i));
8         if a<5
9             m=round(rand*x);
10            if m<1
11                m=1;
12            end
13        else
14            m=sort(unidrnd(a,1,1));
15            b=find(choice(:,i));
16            m=b(m);
17        end
18        if choice(m,i)==0
19            newchoice(m,i)=1;
20        else
21            newchoice(m,i)=0;
22        end
23    end
24 end
25 end
```

## F 异常处理函数

```
1 %用于保证用于拟合插值可以正常进行
2 function [choice]=t(choice,a)
3 for i =1:length(a)
4     m=sort(unidrnd(90,1,3));
5     choice(m,a(i))=1;
6 end
7 end
```

## G 枚举法可能选择

```
1 choice=[];
2 for c1=1:40
3     for c2=c1+5:80
4         for c3=c2+5:90
5             choice=[choice;c1 c2 c3];
6         end
7     end
8 end
```

## H 计算测试集成本

```
1 clear all;
2 a=[ -16  -2   7  52  67];
3 a=a+20;
4 minput=dlmread('dataform_testA.csv');
5 %构造温度、电压矩阵
6 all_temp=minput(1:2:400,:);
7 all_volt=minput(2:2:400,:);
8 p=[];
9 for j =1:200
10     p=[p;spline(all_volt(j,a),all_temp(j,a),all_volt(j,:))];
11 end
12 p=abs(p-all_temp);
13 s=250+(length(find(p>0.5&p<=1))+length(find(p>1&p<=1.5))*5+length(find(p>1.5&p<=2))*10+
    length(find(p>2))*10000)/200;
```