基于神经网络的零件装配约束模型

摘要

本文研究的是利用零件装配约束模型，建立零件与产品质量参数关系的问题。

针对问题一，在产品质量参数模型求解过程中，首先根据**零部件装配约束模型**，为确定**装配约束关系**，检验了数据相关性与分布特征，得到零部件质量参数相互**独立且服从正态分布**，可视为随机变量。为充分利用零部件质量参数的独立性，提出利用 **BP 神经网络**建立数学模型，并利用梯度下降动量算法对网络各节点参数进行优化调整, 通过 MATLAB 编程实现了该模型。根据初始数据对此 BP 网络模型进行训练和测试, 通过测试验证了此模型的有效性。

针对问题二，考虑到辅件参数未知，我们分析了零件装配约束模型并对其进行**线性近似**，由此建立了零件与产品质量参数的**线性回归模型**，利用该模型的拟合误差与辅件质量参数相同的正态分布、无自相关性、对产品标定值影响**相近的数学特性**，**将此误差视为辅件的质量参数**输入到非线性神经网络模型中进行训练，提高了问题一中建立的神经网络模型的精度。

针对问题三，零部件参数设计可归结为在一定条件下求总利润最大的**最优解问题**。我们建立了一个关于零部件质量参数设计的优化模型。利用**蒙特卡洛采样**生成一定数目采样点，输入问题二中神经网络模型得到产品质量参数。通过**模拟退火算法**，得到本问题的解：零部件x_1~x_{10}等级依次为 C、C、C、C、C、B、C、C、C、C，此时产品利润为 865 元,同时证明各零部件均选取最好的等级并非最合理方案。

最后本文对模型进行了**灵敏度分析**，发现此模型能够较好的区分不同零部件质量参数对产品质量参数产生的不同影响，且对辅件参数不敏感，ε的工艺误差不会对最终结果产生明显影响，模型较稳定。

关键词：零件装配约束模型 神经网络 蒙特卡洛采样 模拟退火算法

目录

一、 问题重述

1.1 问题背景

产品零件的参数设计是机械工业设计工作中的重要步骤之一，确定零件等级及参数容差则是参数设计与优化的基本问题。在机械生产过程中，零件质量参数的标定值与容差直接影响了产品的可靠性与稳定性。零件等级选择不合理或容差设计的过大会使产品参数远离目标值，而从经济角度考虑容差设计的过小则会导致产品成本过高。综上，在设计零件参数时需综合考虑产品质量及零件设计成本。

1.2 问题重述

一件产品通常由多个零部件组装而成，产品的质量由零部件质量及生产工艺、生产技术等多种因素决定。某类产品由 10 个主要部件及若干辅件构成。x_i，$i = 1, \cdots 10$ 表示主要零部件的质量参数，y 表示产品的质量参数。产品质量参数主要由主要零部件确定，辅件等的影响一般不超过产品标定值的 5%。请根据上述信息解决下列问题：

（1）不考虑辅件、生产技术等因素的影响，确定产品质量参数与零部件质量参数的关系。

（2）不考虑生产技术等因素的影响，考虑辅件对产品质量的影响，重新回答问题一。

（3）原产品设计方案是各零部件都选最好的等级，评价该方案是否合理？请设计最合理的方案。

二、 问题分析

2.1 问题一分析

该问要求在不考虑辅件、生产技术的前提下，根据附件 2 所给的质量参数统计数据，建立产品质量参数与零件质量参数的非线性关系模型。此问中，我们将辅件和生产工艺的影响忽略至误差中。

由于题设中并未给出具体产品类型以及零件型号，故无法直接确定其关系。为此，我们根据零件装配约束模型，分析零件的装配约束关系；根据零件质量参数相互独立的特点，选用 BP 神经网络进行非线性模型的拟合；针对所给数据符合正态分布的特性，选择贝叶斯正则化算法作为训练函数；通过观察多种隐藏层效果，选取单隐层和 5 个隐藏层节点数以达到最佳拟合效果。

2.2 问题二分析

问题二在问题一的基础上，增加了考虑辅件的要求，并重新建立产品质量参数与零件质量参数间的关系模型。

在附件中，辅件的质量参数为未知，这给直接输入主要零件和辅件的质量参数来拟合模型的方法带来了困难。而根据基于装配约束的部件参数化模型，产品与零件、辅件的参数关系基本可以看作线性模型，故先使用多元线性回归模型对附件二所给的抽样数据进行拟合，由此可以得出线性模

型输出与实际抽样的产品参数间的误差。由于根据部件参数化模型知，使用线性模型拟合理应足够精确，因此可将线性模型拟合所致误差看作辅件影响。

在以上多元线性模型中拟合所得出的误差仅包含了辅件影响，然而若要考虑公差配合、生产工艺等非线性因素的影响，仍须建立非线性模型。我们继续选用神经网络，重新训练模型以改进问题一中未考虑辅件和生产工艺的模型。

2.3 问题三分析

对于问题三，判断各零部件均选取最优等级是否合理，即为确定此方案是否可在产品质量范围内达到利润最大化。求解过程中需综合考虑产品价格、零部件成本、产品及零部件质量参数、辅件对产品质量参数的影响等因素，建立单目标优化模型。我们考虑得到产品利润函数，目标为确定零部件参数x_i使目标函数值达到最大。

本题的求解过程实际上是一种优解搜索过程，由于零部件参数容许范围是一个连续域，穷举法显然是不可行的，从方法的可行性和有效性方面考虑，在满足产品质量要求的条件下，采用蒙特卡洛采样生成一定量采样点，通过模拟退火算法求解零部件参数最优设计。

三、 模型假设

根据零件设计工艺中的一些具体要求，并为达到简化模型的目的，出题目中以给出的假设，我们进一步做了以下假设：

1. 假设在生产单件产品的过程中，每种零件有且只有一个。
2. 假设组成产品的各个零件在生产过程中互不影响，且可以无困难地组装成一件产品。即若视各零件质量参数为随机变量，则它们相互独立。
3. 假设在生产过程中，没有工艺失误造成的产品损坏。
4. 假设各个零件的标定值和容差相互独立。
5. 假设各零件容差的等级与其标定值得比为定值，分别为：A 级 0~±5%，B 级±5%~±10%，C 级±10%~±20%。

四、 符号说明

4.1 符号说明

符号	符号说明
x_i	零部件质量参数
y	产品质量参数
σ	零部件质量参数总体标准差
μ_i	零部件质量参数x_i数学期望
$\rho_{x,y}$	x_i、x_j的威尔逊相关系数
P	产品利润

I	产品售价
C	零部件总成本
C_i	第 i 种零部件成本
r_i	产品利润最大时零件质量参数
ε	辅件质量参数
Q	y 目标值与回归值差值平方和

五、 模型建立与求解

5.1 问题一：产品质量参数的非线性神经网络模型

5.1.1 基于零件装配约束模型的数据分析

本问所使用的参数化设计以约束造型为核心，以尺寸驱动为特征。基于装配约束的部件参数化模型需要在已知各零部件质量参数的基础上引入零件的装配关系函数。

零件之间的常见关系如下图所示。其中，装配约束关系由产品中的实际设计确定，一般而言，零件之间会由于存在公差配合、尺寸干涉等因素的影响而失去独立性。为确定装配约束关系，我们需要对附件提供的零件参数抽样进行独立性检验。

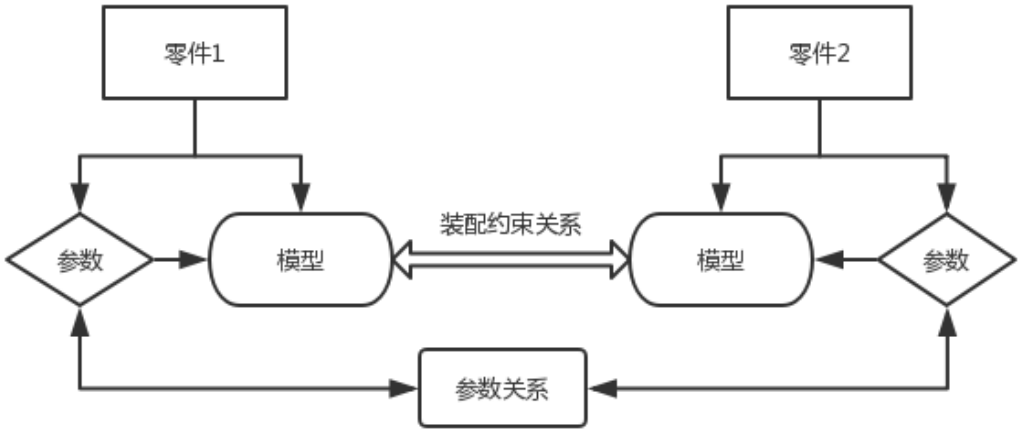

图 1 零件装配约束关系图

零部件质量参数独立性检验

为检验各零件质量参数间独立性以确立装配约束关系，我们建立皮尔逊相关性检验模型，采用 SPSS 中的 Pearson 系数来分析各零件质量参数的关系。

皮尔逊相关也称为积差相关，为英国统计学家皮尔逊于 20 世纪提出的一种计算直线相关的方法，取零部件 1、2 的质量参数x_1，x_2，依据下列公式建立两变量间皮尔逊相关性系数：

$$cov(X,Y) = E\left[(X-\mu_x)(Y-\mu_y)\right]\sigma \tag{1}$$

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_x\sigma_y} = \frac{E(xy)-E(X)E(Y)}{\sqrt{E(X^2)-E(X)^2}\sqrt{E(Y^2)-E(Y)^2}} \tag{2}$$

式中，E 为数学期望， $cov(X,Y)$为指标变量之间的协方差, σ为指标变量的总体标准差, μ 为指标变量的总体均值。

利用 SPSS 对x_1，x_2进行皮尔逊相关性检验，可得相关性分析如下表所示：

表 1 皮尔逊相关性分析

		x1	x2
x1	皮尔逊相关性	1	0.019
	Sig.（双尾）		0.788
	个案数	200	200
x2	皮尔逊相关性	0.019	1
	Sig.（双尾）	0.788	
	个案数	200	200

由上述系列表格的皮尔逊相关系数及显著性分析值，x_1与x_2相互独立。通过以上模型，可分析得到 10 种零部件质量参数x_i，$i=1,\cdots 10$均相互独立，即零件的装配约束关系不在数据中显著体现，可被认为在组装过程中互不影响，可无困难地组成一个产品。

零部件质量参数正态分布检验

经上述分析，10 种零件的质量参数可视为相互独立的随机变量。

为判断参数分布特征，使用 SPSS 对x_1进行分析，正态检验结果如下表所示：

表 2 零部件质量参数正态检验结果

	柯尔莫戈洛夫-斯米诺夫（V）			夏皮洛-威克尔		
	统计	自由度	显著性	统计	自由度	显著性
x_1	0.047	200	0.200	0.993	200	0.420

样本数据为 200 组，数量较小，以柯尔莫哥洛夫—斯米诺夫分析结果为依据，得到数据显著性 $\text{Sig}=0.2>0.05$，符合正态分布。绘制 Q-Q 图如下所示，由图可见基本在直线附近，可以认为服从正态分布：

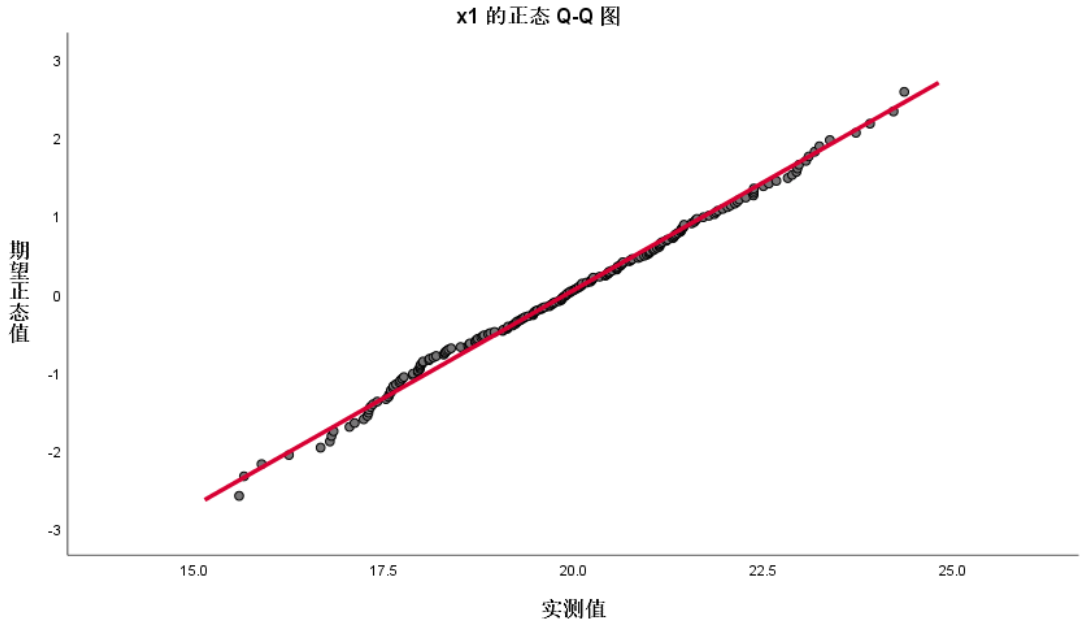

图2 x_1正态分布Q-Q图

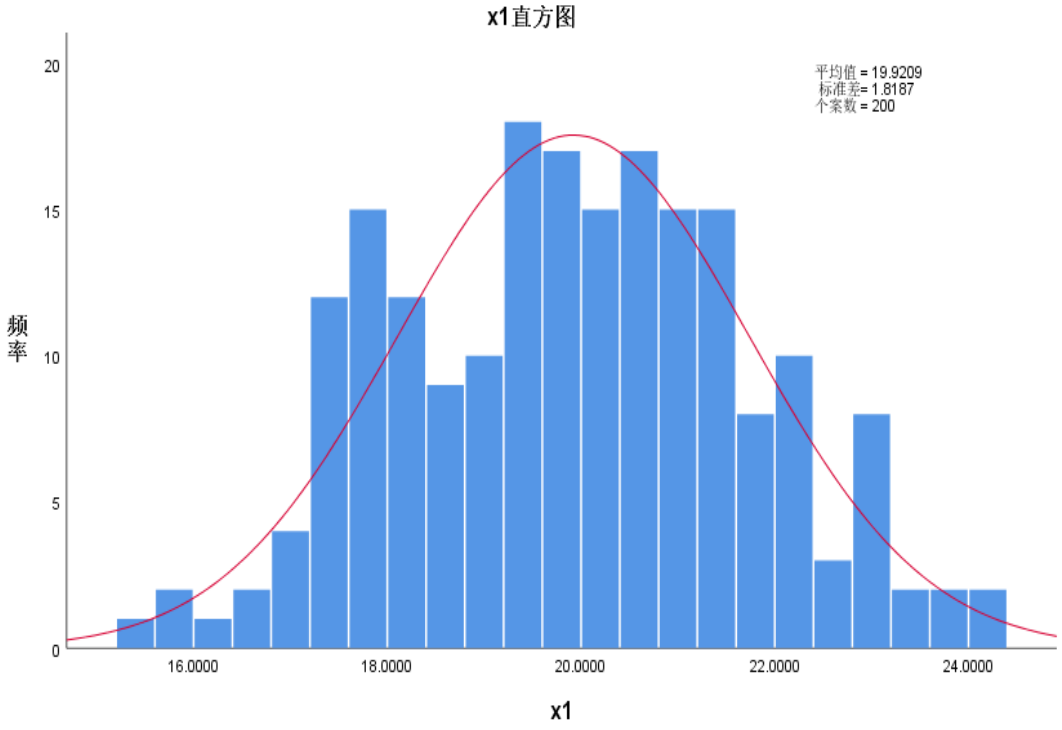

图3 x_1频率分布直方图

根据如下正态分布的密度公式对数据进行拟合：

$$\mathrm{P}(x)=\frac{1}{\sqrt{2+\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}},-\infty<x<+\infty \tag{3}$$

得到x_i，$i=1,\cdots 10$的数学期望μ，方差σ^2如下表所示，可认为，零部件质量参数期望即为标定值。

表 3 零部件质量参数数学期望及方差

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
μ	19.921	19.112	9.784	15.0029	24.949	15.903	18.081	12.311	15.119	22.041
σ^2	3.308	3.808	3.765	3.529	3.439	4.824	4.416	3.837	4.227	3.964

因零部件参数均符合正态分布，可认为数据均为有效数据，无须去除无效数据。

5.1.2 BP 神经网络模型的构建

由于本题所涉及的参数相互独立，我们选用对独立自变量拟合效果最佳的 BP 神经网络来建立非线性模型。

神经网络是一种能够在信息含糊、不完整等复杂环境中处理分析信息的数据驱动建模方法，并且具有自学能力，进而可以对复杂问题做出合理的判断决策，适合用于处理产品质量参数这种动态机理复杂、内部状态未知、仅可获得输入输出数据的复杂工业过程建模。BP 神经网络在本文中的应用原理，就是通过训练样本，采用神经网络进行计算，得出各零件质量参数与产品质量参数之间的数学定量关系。

产品质量参数 BP 神经网络模型的建立，基于以下公式。沿信息传播的方向，给出网络的状态方程。以十种零部件的质量参数作为 BP 神经网络模型的输入，并选用单隐层的 BP 网络结构进行产品质量参数的模拟。网络的各层输入输出关系可描述为：

（1）第一层（输入层）： 将零件的质量参数引入神经网络。

10 个神经元分别为十种零部件的质量参数

$$y_i^{(1)} = x_i \quad i = 1, \cdots 10 \tag{4}$$

其中，x_i为输入层第 i 个神经元的输入；$y_i^{(1)}$为输入层第 i 个神经单元的输出。

（2）第二层（隐含层）：

隐含层包含 N 个神经元：

$$\begin{cases} y_{ij}^{in} = \sum_{i=1}^{n} w_{ij} x_i \\ y_k^{out} = f\left(y_{ij}^{in}\right) \end{cases}, \mathrm{i} = 1, 2, \dots, 10 \tag{5}$$

式中，y_{ij}^{in}为隐含层第 j 个神经元从输入层第 i 个神经元接收到的输入信号，w_{ij}为输入层第 i 个神经元到隐含层第 j 个神经元的权重，y_k^{out}为输出层第 k 个神经元接收到的输出信号。

其中，$\mathrm{f}(x)$为传递函数，可取不同形式，如：

S 函数：

$$f(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

高斯基函数：

$$f(x) = exp\left[\frac{(x-a)^2}{b^2}\right] \tag{7}$$

条基函数，小波函数等，本文采用的是tansig函数，公式如下：

$$\mathrm{tansig}(n) = \frac{2}{1 + \tan^{-1} n} \tag{8}$$

（3）第三层（输出层）：

输出层包含 1 个神经元，为产品质量参数。

$$y_0 = f_1 \sum_j \left(w_{jk} y_k^{out}\right) \tag{9}$$

式中，y_0为输出层神经元的输出信号，w_{jk}为隐含层第 j 个神经元到输出层第 k 个神经元的权重，f_1 为输出层函数。

隐藏层神经元个数对 BP 神经网络预测精度有显著影响，节点数过多，导致训练时间增加，网络过拟合，且往往梯度下降过快；节点数过少，网络不能很好地学习，需增加训练次数，而按乘法更新的误差也易增长过快，训练精度受到影响。我们参考如下公式来确定最适隐藏层神经元个数：

$$\mathrm{l} < \sqrt{(m+n)} + a \tag{10}$$

式中，n 为输入层节点数；m 为输出层节点数；l 为隐层节点数，a 为$[0，10]$区间内的常数。在模型建立过程中，我们通过公式确定隐层节点数的范围，然后通过调参确定最佳节点数。经过试验，我们选择 N=5，此时 BP 神经网络达到了较高的精确度。

训练函数对神经网络的生成有着决定性的作用。针对极佳地符合正态分布的输入数据和较少的样本数量，我们选用贝叶斯正则化(Bayesian regularization)来训练神经网络

基于本文模型，$\mathrm{i} = 10,\ \mathrm{j} = 5,\ \mathrm{k} = 1$,以上步骤完成了神经网络的基本网络构建，神经网络拓扑关系如下图所示：

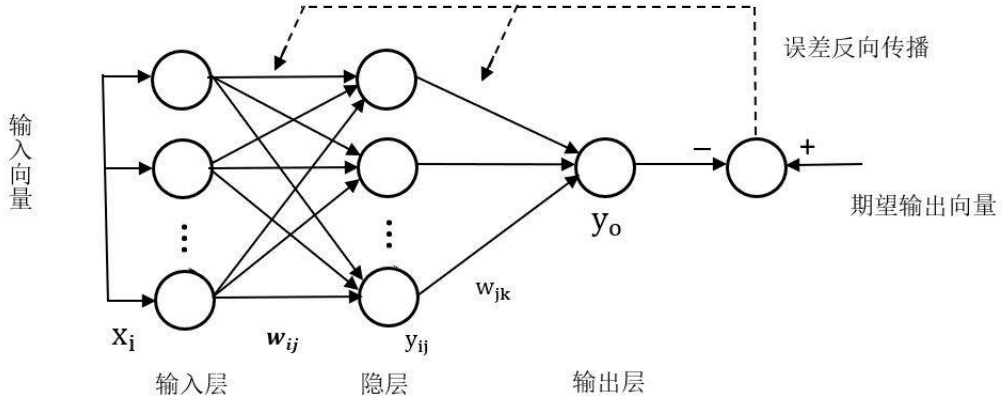

图 4 神经网络拓扑图

输入信号向前传播到隐节点，经过激活函数后，再将隐节点的输出信息传播到输出节点，如果输出层得到的结果与期望的不符，则进入反向传播过程，将误差信号沿原来的路径逐层返回，返回过程修改各层神经元的权值，使误差信号达到最小，最后给出输出结果。

下一步需确定网络的学习算法，调整网络的权值，使网络的实际输出尽可能接近期望输出。

BP 神经网络采用最优化方法中的沿梯度下降算法，目的是使实际输出结果和期望输出结果之间的均方差最小。具体算法如下：

当网络输出质量参数与期望输出质量参数不等时，存在输出误差 E 为:

$$\mathrm{E}=\frac{1}{2}(y_o-y)^2 \tag{11}$$

其中，y_o表示产品质量参数网络实际输出值，y 表示产品质量参数实际输出值。

引入动量因子 η，得到各值间连接权值或阈值的调整算法如下：

$$\Delta \mathrm{W}(k+1)=\eta\Delta \mathrm{W}(k)+\alpha\big(1-\ \eta\ \big)\frac{\partial E(k)}{\partial W(k)} \tag{12}$$

$$\mathrm{W}(k+1)=W(k)+\Delta W(k+1) \tag{13}$$

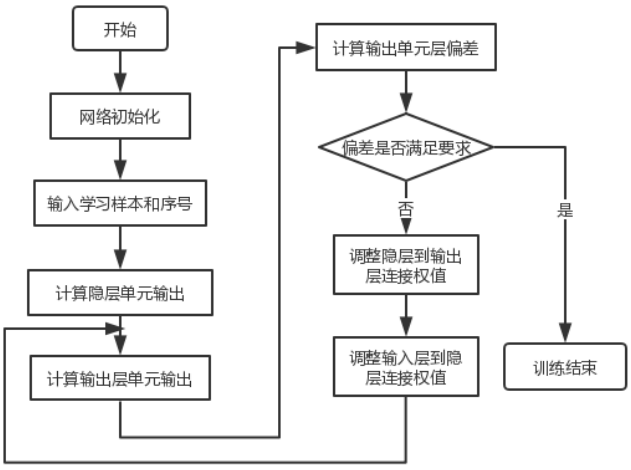

图 5 神经网络训练流程图

5.1.3 BP 神经网络模型的求解

数据分析与预处理

为方便后续数据处理，使最终数据更加客观反应所代表因素的实际意义，并保证程序运行时收敛加快，对样本数据进行归一化处理，将原始数据压缩到[0,1]闭区间，有利于神经网络的训练。我们将数据集按如下公式进行归一化处理：

$$x_j = \frac{x_{io} - x_{min}}{x_{max} - x_{min}} \tag{14}$$

式中，x_j为归一化处理后数据、x_{io}为初始数据、x_{max}, x_{min}分别为初始数据中最大值与最小值。

BP 网络的建立与训练

为得出各零件质量参数与产品质量参数之间的数学定量关系，我们建立 10-5-1 的 BP 网络结构，其中 10 表示输入项，5 为隐藏层神经元个数，1 为输出项。结构图如下：

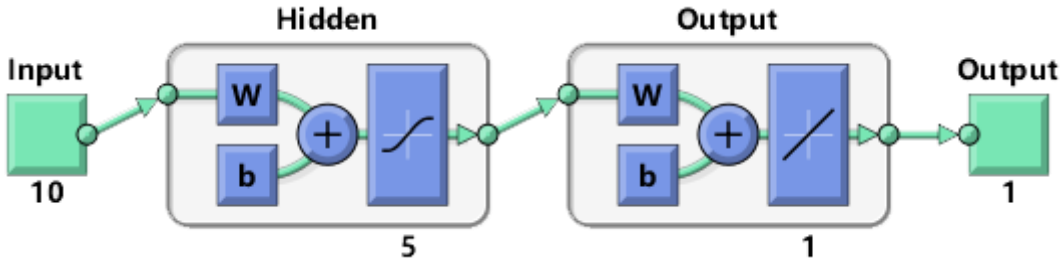

图 6 BP 网络结构图

上述模型训练结果如下图所示：

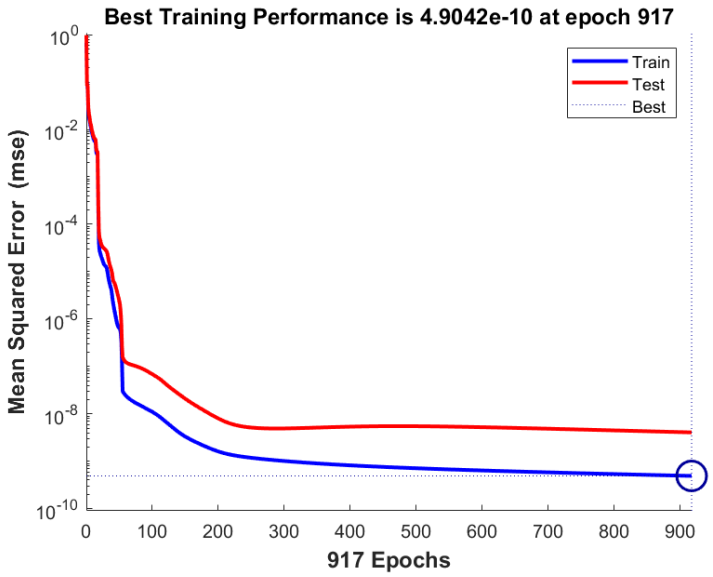

图 7 BP 网络误差曲线图

分析误差曲线图，可以看出该模型的训练误差下降较快，预测值基本跟踪实际值。在计算资源充足且对精度要求比较高的情况下，训练步数越大越好。

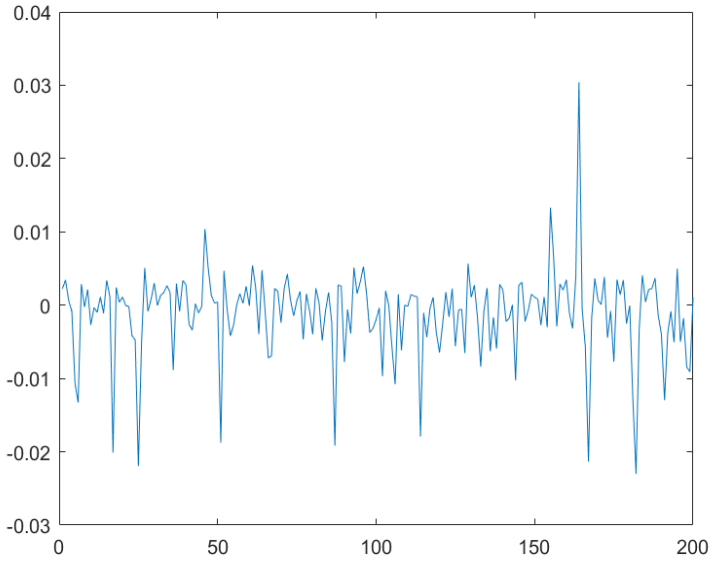

图 8 实际输出和预测输出对比图

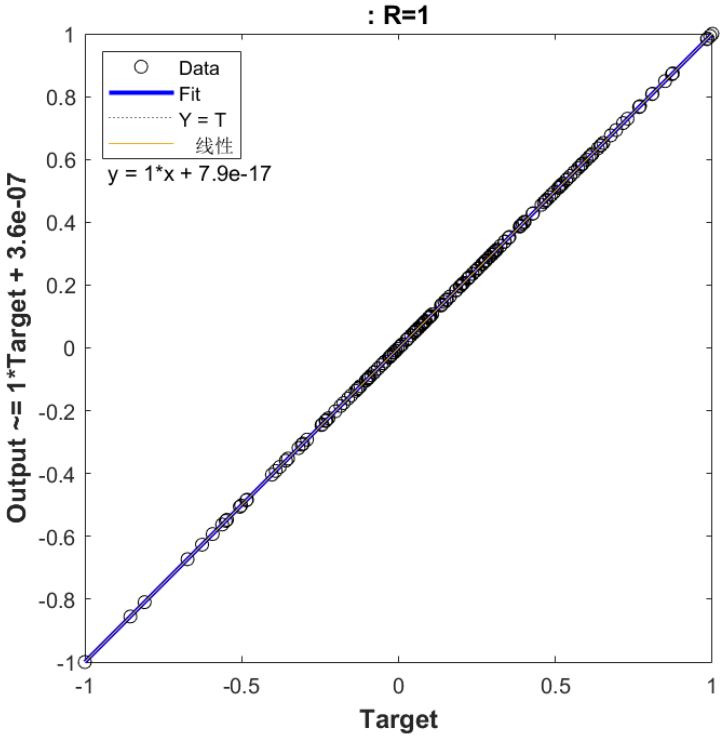

图 9 网络训练结果回归图

分析网络训练结果回归图可知，R 值为 1，说明通过神经网络建模，可以较好拟合产品质量参数与零部件质量参数的非线性关系模型，且拟合精度可以满足生产要求。

该模型的均方误差为 10^{-9} 级。

产品质量参数的 BP 神经网络模型求解结果

用权值和阈值表示 BP 神经网络建立的产品质量参数数学模型结构, 权值和阈值表示如下:

（1） 输入层与隐层间权值:

表 4 BP 网络输入与隐层间权值

-0.0615	-0.0683	-0.0759	-0.0766	-0.0860	-0.0890	-0.0739	-0.0813	-0.0789	-0.0940
-0.0359	-0.0379	-0.0409	-0.0413	-0.1071	-0.0474	-0.0388	-0.0466	-0.0417	-0.0087
-0.0132	-0.0136	-0.0144	-0.0144	-0.0452	-0.0166	-0.0135	-0.0169	-0.0145	-0.1310
-0.0613	-0.0689	-0.0772	-0.0780	-0.0956	-0.0907	-0.0756	-0.0814	-0.0807	-0.0994
0.0186	0.0194	0.02055	0.02064	0.1153	0.0237	0.0193	0.0240	0.0207	0.0636

（2）隐层阈值：

$[-0.7110550219 \quad 0.6529422763 \quad 0.6619843315 \quad 0.6137957060 \quad 0.6390429999]$

（3）隐层与输出层间权值：

$[-2.7022162568 \quad -0.83759284147 \quad -2.2451893496 \quad -2.1874433034 \quad 1.5614096337]$

（4）输出层阈值：2.39240168811338

5.2 问题二：考虑线性辅件和生产工艺影响的非线性神经网络模型

5.2.1 对零件装配约束模型的线性近似

该问要求计及辅件对产品质量的影响，然而辅件的质量参数并未直接给出，因此考虑使用近似模型先行确定辅件的质量参数。以下对部件装配模型进行近似操作。

部件装配模型的结构为如下图所示的树状结构，产品模型由若干子装配模型及零件组成，而各子装配模型又由多个子装配模型和零件装配而成。

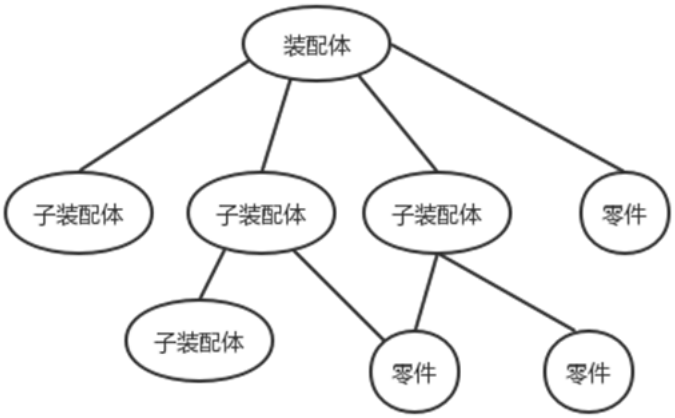

图 10 产品装配图

由于不考虑生产技术，且装配关系已在 5.1.1 节中确定未在数据中显著体现，子装配体可被近似为装配过程，装配体从而近似为零件的线性组合，如下图所示。

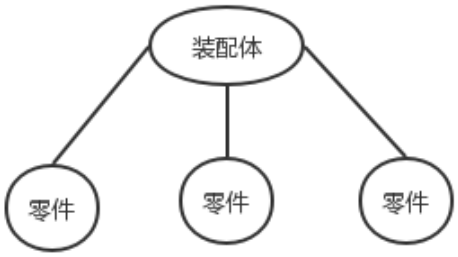

图 11 产品近似装配图

其中，辅件虽然作为产品的线性零件，但对产品质量参数影响较小，不超过产品标定值的 5%。而同样作为零件，我们可以假设辅件的质量参数与题设所提供的主要零部件的特性相同，即符合正态分布、无自相关性。

综上，辅件可认为具有正态分布、无自相关性的特征，与线性回归模型中的随机误差项非常相近。因此，我们选用多元线性回归来拟合产品的质量参数，将其拟合误差作为辅件的质量参数。

5.2.2 基于近似线性模型的辅件质量参数的确定

主要零件和产品的第 m 次抽样的质量参数为：

$$[x_{1m} \quad x_{2m} \quad \cdots \quad x_{10m}]\ ,\ y_m\ ,\ m = 1,2,\ldots,200 \tag{15}$$

则其线性关系为：

$$\begin{cases} y_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_{10} x_{110} + \varepsilon_1 \\ y_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \cdots + \beta_{10} x_{210} + \varepsilon_2 \\ \vdots \\ y_{200} = \beta_0 + \beta_1 x_{2001} + \beta_2 x_{2002} + \cdots + \beta_{10} x_{20010} + \varepsilon_{200} \end{cases} \tag{16}$$

其中$\beta_0, \beta_1, \ldots, \beta_{10}$为 11 个待估计参数，$\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{200}$为 200 个相互独立且服从同一正态分布$N(0,\sigma)$的随机变量。

令

$$\mathrm{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{200} \end{pmatrix} \tag{17}$$

$$\mathrm{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{110} \\ 1 & x_{21} & x_{22} & \cdots & x_{210} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{2001} & x_{2002} & \cdots & x_{20010} \end{pmatrix} \tag{18}$$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{10} \end{pmatrix} \tag{19}$$

$$\mathrm{E} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_{200} \end{pmatrix} \tag{20}$$

则公式*(1)*可重写为

$$Y = XB + E \tag{21}$$

为了估计参数 β，我们采用最小二乘法。设$b_0, b_1, \ldots, b_{10}$分别是参数$\beta_0, \beta_1, \ldots, \beta_{10}$的最小二乘估计，则回归方程为：

$$\hat{y} = b_0 + b_1 x_1 + \ldots + b_{10} x_{10} \tag{22}$$

由最小二乘法可知，$b_0, b_1, \ldots, b_{10}$应使目标值y_m与回归值$\hat{y}_m$的差值平方和 Q 达到最小，而 Q 为非负平方项，定有最小值。根据微积分中的极值定理，$b_0, b_1, \ldots, b_{10}$应为以下方程组的解：

$$\begin{cases} \dfrac{\partial}{\partial b_0} Q = -2 \sum\limits_m (y_m - \hat{y}_m) = 0 \\ \dfrac{\partial}{\partial b_i} Q = -2 \sum\limits_m (y_m - \hat{y}_m) x_{mi} = 0 \end{cases} \tag{23}$$

显然，该方程组的系数矩阵为对称矩阵，记为 A，则$A = X'X$；常数项矩阵记为 C，可知$C = X'Y$。故回归方程的回归系数为：

$$b = A^{-1}B = (X'X)^{-1}X'Y \tag{24}$$

在 MATLAB 中运用上述公式进行编程，可得回归系数为：

$$b = [-10.8323 \quad 1.0041 \quad 1.0043 \quad 0.9961 \quad 1.0013 \quad 1.4926 \quad 0.9958 \quad 0.9949 \quad 1.0010 \quad 0.9993 \quad 1.4403]$$

回归方程为

$$\begin{aligned} \hat{y} = (&-10.8323 + 1.0041x_1 + 1.0043x_2 + 0.9961x_3 + 1.0013x_4 + 1.4926x_5 \\ &+0.9958x_6 + 0.9949x_7 + 1.0010x_8 + 0.9993x_9 + 1.4403x_{10}) \end{aligned} \tag{25}$$

则得误差$\varepsilon = \hat{y} - y$，即认为是辅件的质量参数。

我们对该线性模型进行残次误差分析，由如下残次误差图可知，尽管大多数输出值与目标值在可接受的误差内，但仍有不少点存在较大误差，且均为正向误差，因此，该线性模型性能并不理想，需要使用更加精确的模型来拟合。但是作为用于输出误差作辅件质量参数的模型，其性能基本可以接受。

图 12 模型残次误差图

使用公式$\sum \frac{|\varepsilon|}{y}$对辅件对产品的影响进行评估，得到 5.38%的结果。考虑到线性近似模型中被忽略的装配关系，该结果可以认为符合题设的一般不超过标定值的 5%的设定。

5.2.3 包含辅件质量参数的神经网络模型的构建和求解

在取得上述辅件的质量参数后，我们重新建立了 BP 神经网络以确立考虑辅件的各零件质量参数与产品质量参数关系模型。该模型为 10-5-1 的 BP 网络结构，拓扑结构图如下：

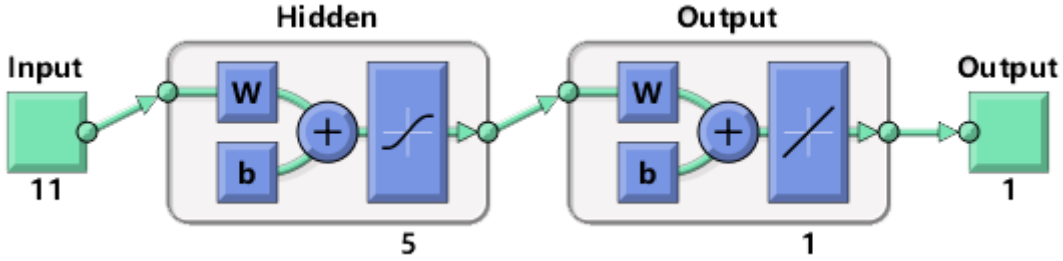

图 13 BP 网络结构图

上述模型训练结果如下图所示：

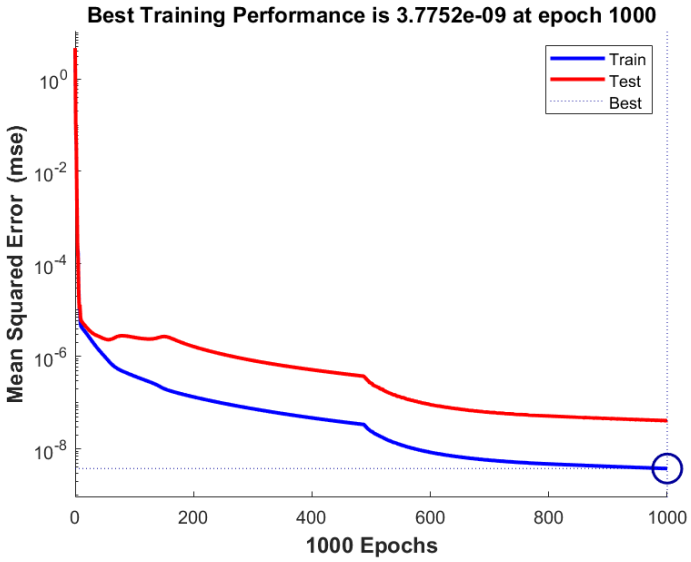

图 14 BP 网络误差曲线图

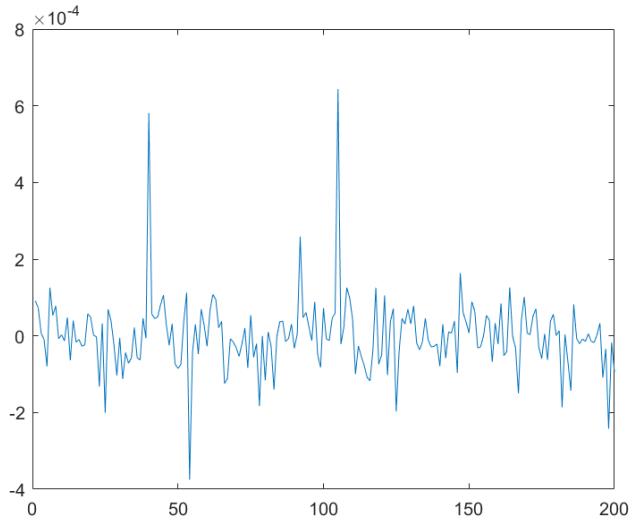

图 15 实际输出和预测输出对比图

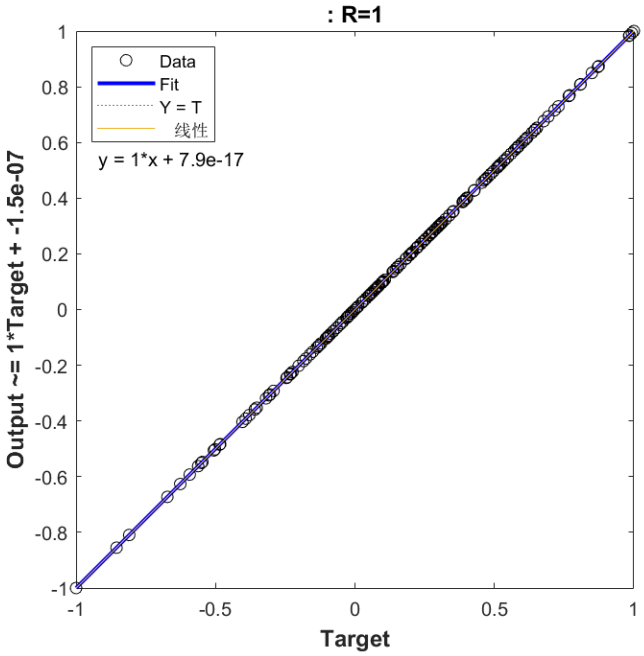

图 16 网络训练结果回归图

该模型的均方误差为 10^{-12} 级，与问题一所生成的模型相比，均方误差下降了 10^{3}，准确性得到了一定的提升。

用权值和阈值表示 BP 神经网络建立的产品质量参数数学模型结构, 权值和阈值表示如下:

（1）输入层与隐层间权值:

表 5 BP 网络输入与隐层间权值

0.0406	0.0440	0.0476	0.0497	0.0651	0.0550	0.0497	0.0549	0.0508	0.0718	0.0013
0.0669	0.0737	0.0804	0.0823	0.1091	0.0946	0.0778	0.0869	0.0843	0.1151	-0.0022
-0.0669	-0.0736	-0.0804	-0.0820	-0.1085	-0.0942	-0.0782	-0.0876	-0.0840	-0.1155	0.0021
-0.0474	-0.0523	-0.0562	-0.0601	-0.0804	-0.0684	-0.0547	-0.0595	-0.0613	-0.0808	0.0009
0.0388	0.0429	0.0488	0.0449	0.0590	0.0546	0.0441	0.0528	0.0465	0.0666	-0.0048

（2）隐层阈值：

$[-0.074809762772 \quad -1.0842609545 \quad -1.1317141156 \quad -0.065841715120 \quad -0.0080242547192]$

（3）隐层与输出层间权值：

$[1.170483831220261.64739675430338-1.68757054111852-1.540108837257551.54705974600291]$

（4）输出层阈值：0.131240057746703

5.3 问题三：基于模拟退火算法的蒙特卡洛抽样零部件等级选取方案

判断各零部件均选取最优等级是否合理，即为确定此方案是否可在产品质量范围内达到利润最大化，我们将其转化为最优问题。建立描述产品利润的目标函数，并求取最优解。

首先采用遍历搜索方式，此算法时间复杂度较高，于是我们建立基于蒙特卡洛和模拟退火算法的最优模型来求解此问题。

5.3.1 零部件等级对产品等级影响分析

每个零部件等级的提升对整体等级的提升有正面影响，但由于零部件售价随着零部件的等级提升而提升，故各个零部件均采用最优等级未必能使总体利润最大化。面对最优化问题，利用问题二生成的神经网络模型，结合数据我们可以采用一种用于求最优解的概率算法，也就是模拟退火算法

5.3.2 遍历法问题分析

基于对问题二神经网络模型的分析，零部件的质量提升在一定程度上提升了产品整体质量，但由于不同等级零件的成本差异，零件等级最优时利润未必是最优解。由于零件等级取值与产品整体等级取值均有限，故可能得到的利润是有限的。

在每个等级允许范围内取最接近最优等级的质量参数，由于质量零件参数提升会在一定程度上提升产品质量，可以取得相应等级内最大利润。相同等级下同一零件售价相同，可得出产品生产成本，进而得出产品利润。因此遍历每种零件的所有可能等级，并记录最大利润对应的零件等级选择则可求解此最优化问题。

时间复杂度：$O(m^n)$（m 为零件等级取值，n 为零件个数）

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} years
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years	very long
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years	very long
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

数据来源：Princeton University本科算法兴趣小组实验

图 17 遍历算法时间复杂度

上图为遍历算法时间复杂度数据，对于 5 种等级范围，10 个零件的问题，遍历算法需要迭代 9765625 次，根据时间复杂度与求解时间表格估计所需运行时间为十分钟左右。对于小规模输入此算法有一定的可行性，但当零件数目与可能等级范围增加时，此算法的时间开销将变得不可取，于是我们利用一类基于概率的蒙特卡洛和模拟退火算法来求解此最优化问题

5.3.3 基于蒙特卡洛和模拟退火算法的模型建立

考虑实际生产销售过程中，在确定各零件等级时，需提高获得高利润的概率，引入蒙特卡洛法生成随机取样点，使点集的置信区间值更接近于实际情况。

蒙特卡洛法，也就是随机试验点法。它的基本思想是：在函数的可行域内随机地选取实验点，由于随机取得的点在区域中分配比较均匀，所以对函数的大致形态能较好的体现。

零件$x_1 \sim x_{10}$的质量参数均符合正态分布，可通过蒙特卡洛算法随机生成每一种零件的质量参数。本问模型中的随机点由以下方法产生：

$$\mathrm{data}_{\mathrm{pool_i}} = \mathrm{normrnd}\left(\mathrm{average}_{\mathrm{x(i)}}, \mathrm{stanard}_{\mathrm{x(i)}}, 100\right) \tag{26}$$

生成随机散点图如下图所示：

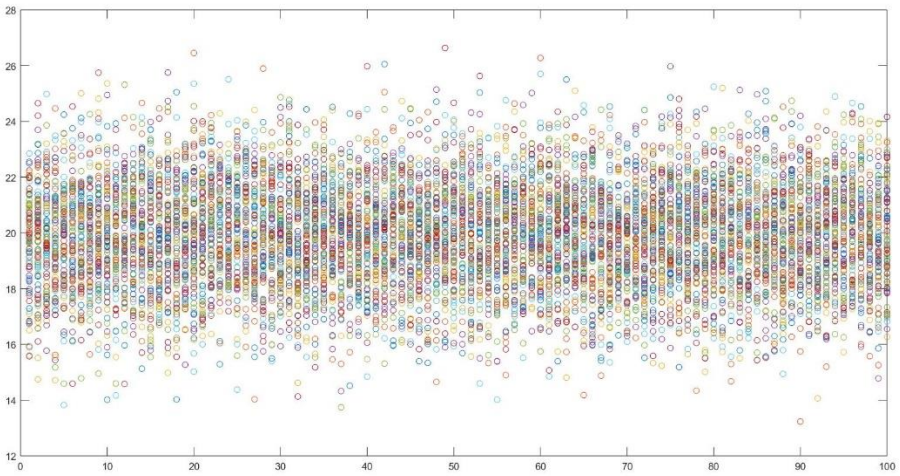

图 18 蒙特卡洛随机采样图

根据问题二中得到的神经网络模型，将采样得到的离散点作为网络的输入得到产品质量参数输出，采用模拟退火算法求取最优解。模拟退火算法流程图如下图所示：

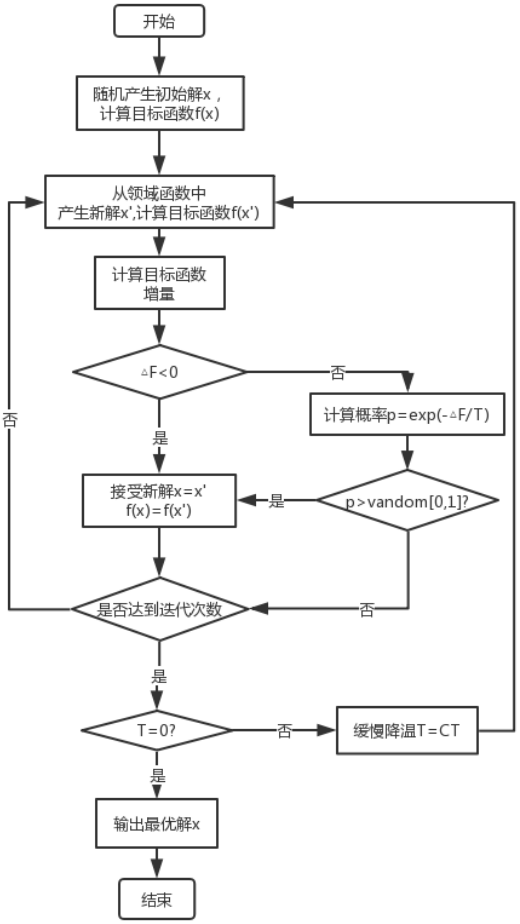

图 19 模拟退火流程图

模拟退火算法源于固体退火原理，规定优化问题的一个解i及目标函数f(i)，算法持续进行“产生新解—判断—接受/舍弃”的迭代过程，即执行 Metropolis 过程。

Metropolis 准则对应的转移概率P_i如下式所示：

$$P_i(i \to j) = \begin{cases} 1, & f(j) \le f(i) \\ Exp\left(\dfrac{f(i) - f(j)}{t}\right), & f(j) > f(i) \end{cases} \tag{27}$$

模拟退火算法依据上式接受新的解，因此除接受优化解外，还在一定范围内接受恶化解，既可以从局部最优的陷阱中跳出，更有可能求得组合优化问题的最优解。

采用模拟退火算法，包含控制参数 T，退火系数 a,以及目标函数（即就是利润最大）max，其基本过程如下。由于零件的等级依据其相对于标定值的距离进行划分

（1）对于标定值，B 类区间，C 类区间：

$$\{\mathrm{A}, B_1, B_2, \mathrm{C}_1, C_2\} \tag{28}$$

对于A类零件，规定：

$$\left|\frac{x_i - \mu_i}{\mu_i}\right| \le 0.05 \tag{29}$$

对于B_1类零部件，规定：

$$-0.1 \le \frac{x_i - \mu_i}{\mu_i} \le -0.05 \tag{30}$$

对于B_2类零部件，规定：

$$0.05 \le \frac{x_i - \mu_i}{\mu_i} \le 0.01 \tag{31}$$

对于C类零部件，规定：

$$-0.2 \le \frac{x_i - \mu_i}{\mu_i} \le -0.1 \tag{32}$$

对于C_2类零部件，规定：

$$0.1 \le \frac{x_i - \mu_i}{\mu_i} \le 0.2 \tag{33}$$

其中，x_i为零部件质量参数，μ_i为该种零部件质量参数标定值。

基于 5.1.2 中对零部件质量参数的正态分布检验，x_i均符合正态分布，可视x_i数学期望为相应零部件标定值。实际编码时采用$[-2,2]$的整数区间取值表示五个等级。模拟退火算法首先起始于零件等级的某个随机状态，零件$x_1 \sim x_{10}$分别随机地取一定的等级，之后退火过程分为多个部分：随机改变状态的 next 步骤，评估新状态并给出对应的权值的评估步骤，判断新状态是否优于旧状态的裁决部分步骤，当我们确定一个更优值时我们进入降温 dec 部分。

（2）目标函数的建立：

判断各零部件均选取最优等级是否合理，即为确定此方案是否可在产品质量范围内获得最大利润。求解过程中需建立单目标优化模型。我们考虑将产品利润函数作为本题目标函数，影响产品利润的因素如下图所示：

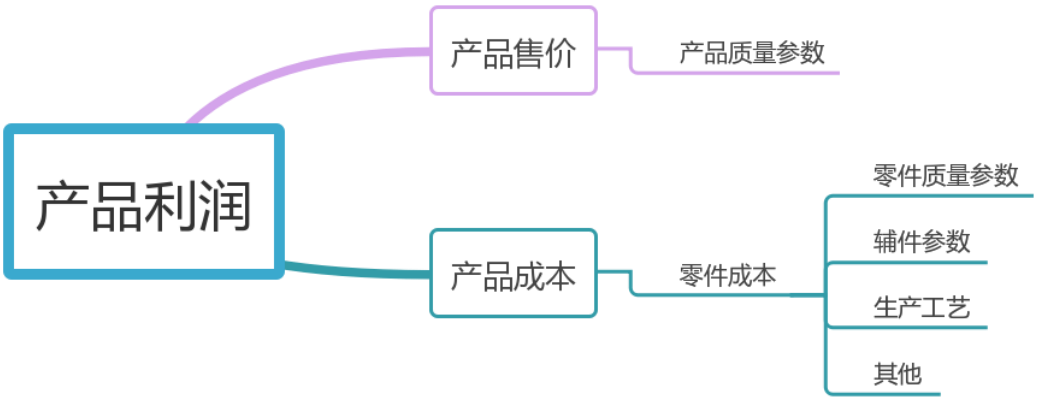

图 20 影响产品利润的因素

零部件成本由其容差决定，规定零部件总成本 C：

$$\mathrm{C}=\sum_{i=1}^{10} C_i=\sum_{i=1}^{10} x_i[abs(x_i)+1] \tag{34}$$

其中，C_i表示第 i 种零件成本。

产品的利润 P 受两部分因素影响：产品售价 I 及零件总成本 C，即：

$$\mathrm{P}=\mathrm{I}-\mathrm{C}=\mathrm{I}-\sum_{i=1}^{10} x_i[abs(x_i)+1] \tag{35}$$

由于使用$[-2,2]$间的整数表示零件等级，可以直接利用$|\mathrm{x_i}|+1$来确定第 i 个零件成本（例如第一个零件等级为 A 时，$C_i=\mathrm{x}_1[abs(0)+1]=100$）

模拟退火算法在温度高于阈值的情况下，环境处于不稳定状态，由 next 步骤在原状态上发生变更，生成新的状态。

（3）更新零部件等级，采用变化法改变零件x_1~x_{10}的等级，并确定零件质量参数

$$x_i=\mathrm{average}_{\mathrm{x(i)}}*(1+0.05*x_i) \tag{36}$$

若此时产品利润高于已知最优值，则将当前利润作为最大利润并记录此时的零件等级，否则进行一定概率的替换，即若此时产品利润低于已知最优值，维持原本状态。无论是否发生变化，系统必定趋于稳定，当状态满足阈值要求时停止循环，此时的已知最优值便是全局最优值。

5.3.4 零部件等级最优方案求解

利用模拟退火算法，我们首先选择零件x_1~x_{10}的等级以此确定生产成本 C，并利用问题二的模型计算得出此时产品质量参数 y 并以此确定所能获取的利润，利用$\mathrm{P}=I-C$得出在该组零件等级下所能获得的利润，为记录历史最优方案并与当前方案进行比较，记录已知最大利润history_max与取得最大利润时零件等级$\mathrm{r_1}$~r_{10}，将 P 与P_{max}比较保留大者并更新零件等级，公式如下：

$$P_{max}=\begin{cases}P\ ,P>P_{max}\\P_{max}\ ,P<P_{max}\end{cases} \tag{37}$$

$$r_i=\begin{cases}x_i\ ,P>P_{max}\\r_i\ ,P<P_{max}\end{cases} \tag{38}$$

式中，P 为当前产品利润，P_{max}为历史产品最大利润。

在每次迭代结束后更新控制参数$T = a * T$，当控制数不满足阈值时停止迭代。

最优方案如下：

表 6　零部件等级最优方案

零件种类	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
等级	C	C	C	C	C	B	C	C	C	C

此时产品预期利润为 865 元

分析上表结果可知，本模型得到了产品利润最大化的零部件等级选取方案，同时证明原方案中所有零部件均选择最好的等级并非最合理方案。

5.3.5 基于模拟退火的理想化方案分析

5.3.4 中提出了最优解模型，并得到最大概率获得产品高利润的零部件等级方案。考虑到上述方法通过比较每种等级方案下的平均利润，从而确定最优解，弱化小概率事件，我们利用模拟退火算法在全局范围内寻找最优解，结果如下表所示：

表 7 理想化零部件等级方案

零件种类	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
等级	C	C	C	C	C	C	C	C	C	C

此时产品利润为 905 元。

分析上述结果，在极端情况下，当各零部件均选用 C 等级时，有小概率获得质量参数较高的产品，从而获得较高利润。而考虑到实际生产过程中，若所有零部件等级均选取 C 等，会导致大部分产品质量参数低，销售价格低，不符合生产与销售需求。

因此，考虑对比不同零部件等级配置，使获得高利润的概率达到最大，以上理想化模型存在一定不合理性。与 5.3.4 中结果进行对比，证明基于蒙特卡洛和模拟退火算法的模型可以最大概率获得最高利润。

六、 模型检验

6.1 问题一模型对各零部件质量参数的灵敏性

考虑到实际机械生产过程中，由于做工及测量精度等问题，各零部件的质量参数可能存在一定偏差，有必要分析问题一中模型对零部件质量参数的敏感性。分别考虑零部件质量参数x_i，$i = 1, \cdots 10$厚度变化对产品质量参数的影响。对各零部件质量参数先后进行±5%范围内的调整，得到产品质量参数与零部件质量参数变化关系如图：

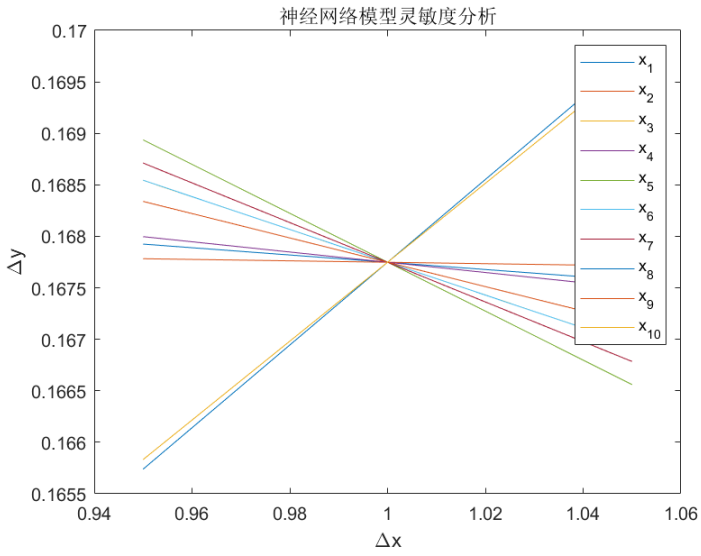

图 21 零部件对产品质量参数的灵敏性

可以看出，零部件x_1，x_3的质量参数对产品质量参数影响较大，说明我们建立的模型对零部件1、3较为敏感，而且能够较好的区分不同零部件质量参数在实际生产过程中对产品质量参数产生的不同影响。

6.2 问题二模型对辅件参数的灵敏性

在我们建立的模型中，辅件对产品质量参数的影响根据神经网络模型线性化后误差得到，研究模型对辅件参数的灵敏性是从侧面研究结果的可靠性。

通过研究辅件参数对产品质量参数的影响来进行灵敏性分析，结果如下图所示：

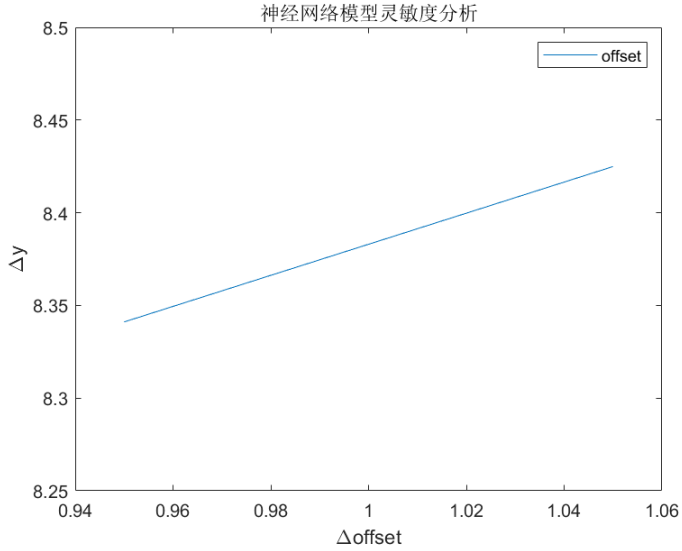

图 22 辅件对产品质量参数灵敏性

可以看出，辅件参数ε的小范围变化对产品质量参数影响不大，说明我们的模型对辅件参数不敏感，ε的工艺误差不会对最终结果产生明显影响，模型稳定性较好。

七、 模型推广

在模型的建立过程中，我们运用概率论、神经网络、模拟退火等知识，简洁地对实际问题构造了一种数学模型，该模型可用于一般的产品设计与零件参数选择，其给出的目标函数也可以用于通常的产品生产中以估算利润。在建模过程中，我们充分发挥了计算机功能，针对求解灵活地调整程序，大大提高了运行的效率。综上，本模型可广泛用于实际生产中的诸多领域，对受多种困难影响的具体问题可以很方便地求出优化解，而且可以用于理论分析和科学实验中。

八、 模型评价与改进

8.1 模型优点

（1）问题一中，由于产品质量参数分布受多种因素影响，很难精确给出各零件质量参数共同作用下的公式。而神经网络具有较强的自组织、自适应与自学习能力，很好地解决了这一问题。神经网络模型能够在未完全了解产品质量参数确定机理的情况下，完成自变量与因变量之间的非线性规划。

（2）问题一中，在标准 BP 算法的基础上，引入动量因子, 对 BP 算法进行改进, 从而使 BP 算法的迭代次数变少,收敛效果变好，并减小了误差, 最终提升网络拟合能力, 改善了 BP 算法的品质。

（3）对于问题三，模拟退火算法在优化零部件质量参数时计算过程简单、通用、鲁棒性强，可用于求解复杂的非线性优化问题。如果降温过程足够缓慢，得到的解的性能会较好。

8.2 模型缺点

（1）问题一、二中，由于神经网络构建所需数据量较大，而题目中给出的所有数据只有 200 组，无法更加精确地拟合出产品质量参数

（2）问题三中，尽管理论上只要计算时间足够长, 模拟退火法就可以保证以概率 1.0 收敛于全局最优点。但是在实际算法的实现过程中, 由于计算速度和时间的限制, 在优化效果和计算时间之间存在矛盾, 难以保证计算结果

8.3 模型的改进方案

在问题一、二的神经网络模型中，可考虑将遗传算法与神经网络相结合，从网络拓扑结构及参数选择两个角度对模型进行优化。即预先固定好网络的拓扑，随后用遗传算法优化网络的权重，以克服 BP 神经网络局部寻优与收敛缓慢的缺点。

对于问题三，今后可对产品质量参数模型的性质做进一步分析，以减小搜索的范围。在搜索的方法上，可采用遗传算法，对染色体的基因组浮点编码，通过繁殖交叉从而在大量解空间内很快地接近全局最优解。若不考虑生产工艺上的影响，产品质量参数模型具有连续性，基因编码的方式即可行。

在实际生产过程中，影响因素将更为复杂，例如零部件质量参数的变化可能导致产品无法良好装配，对产品的质量也会有一定程度上的要求。所以在今后的研究中，应适当考虑模型的效率、工艺的合理性及简洁性及人事等诸多方面的因素。

参考文献

[1] 陈酋滔. 零部件参数化设计方法研究与系统实现[D].南京理工大学,2004.

[2] 周密,何均,姜发玉.零件参数设计模型[J].杭州电子工业学院学报,1997(04):74-81.

[3] 林盾,陈俐.BP 神经网络在模拟非线性系统输出中的应用[J].武汉理工大学学报(交通科学与工程版),2003(05):731-734.

[4] 宋晶晶. 产品技术特征与零部件特征配置及其优化技术研究[D].长安大学,2011.

[5] 李温鹏, 周平. 高炉铁水质量鲁棒正则化随机权神经网络建模. 自动化学报, 201X, XX(X): X−X

[6] Zhou P, Guo D W, Wang H, Chai T Y. Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. IEEE Trans. Neural Networks and Learning Systems, to be published

附录

问题一神经网络模型：

```
% 导入数据
B2 = xlsread('2019B 附件.xlsx','附件 2');
x=B2(:, 1:10);
y=B2(:, 12);

% 正态分布检测，例如：
jbtest(x(:,1)) % 检测 x1 是否呈正态分布，0 表示非常符合正态分布

% 相关性检测，例如：
corr(x(:,1), x(:,2)) % 检测 x1 与 x2 是否相关，越接近 0 表示越不相关

% 归一化，不归一化的话后面就会因为梯度过早达到阈值而使得误差过大
[xn, PSx1] = mapminmax(x');
yn = mapminmax(y');

% 建立神经网络
net = feedforwardnet([5],'trainbr');
net = train(net, xn, yn); % 可能需要执行多次，若迭代次数为 1000，则需要执行下一次

fprintf('问题一的神经网络均方误差为%e\n',mse(net, yn, sim(net, xn)));
```

灵敏度分析一：

```
xm = mean(xn,2);
for i = 1:1:10
    xl = xm*ones(1,11);
    xl(i, :) = xl(i, :) .* [0.95:0.01:1.05];
    plot(0.95:0.01:1.05, sim(net, xl));
    if i == 1
        hold;
    end
end
title('神经网络模型灵敏度分析');
xlabel('{\Delta}x');
ylabel('{\Delta}y');
legend('x_1','x_2','x_3','x_4','x_5','x_6','x_7','x_8','x_9','x_{10}');
```

问题二神经网络模型：

```
% 导入数据
B2 = xlsread('2019B 附件.xlsx','附件 2');
x=B2(:, 1:10);
y=B2(:, 12);

% 多元线性回归
[b, bint, r, rint, stats] = regress(y, [ones(200,1),x]);
rcoplot(r,rint) % 输出残差图

% 正态分布检验
jbtest([ones(200,1),x] * b - y)

% 归一化，不归一化的话后面就会因为梯度过早达到阈值而使得误差过大
x2 = [x, [ones(200,1),x] * b - y];
[xn2, PSx2] = mapminmax(x2');
[yn, PSy] = mapminmax(y');

% 建立神经网络
net2 = feedforwardnet([5],'trainbr');
net2 = train(net2, xn2, yn); % 可能需要执行多次，若迭代次数为 1000，则需要执行下一次

fprintf('问题二的神经网络均方误差为%e\n',mse(net2, yn, sim(net2, xn2)));
```

灵敏度分析二：

```
om = mean([x'; ([ones(200,1),x] * b - y)'], 2);
ol = om*ones(1,11);
ol(11, :) = ol(11, :) .* [0.95:0.01:1.05];
on = mapminmax(ol);
plot(0.95:0.01:1.05, [0.995:0.001:1.005] * 8.383);
title('神经网络模型灵敏度分析');
xlabel('{\Delta}offset');
ylabel('{\Delta}y');
ylim([8.25,8.5]);
legend('offset');
```

并行运算的蒙特卡洛采样与模拟退火算法：

```
%读入表格，并从表格中读入 x,y
B1 = xlsread('2019B 附件.xlsx','附件 1');
B2 = xlsread('2019B 附件.xlsx','附件 2');

x=B2(:, 1:10);
y=B2(:, 12);

CoreNum=8;
parpool(CoreNum);

%测试数据 x,y 是否满足正态分布
if jbtest(x(:,1))==0
    disp("零件参数数据符合正态分布")
end
if jbtest(y(:,1))==0
    disp("产品质量参数数据符合正态分布")
end

%相关性检测，检测每个零件间的相关性
for i=1:9
    corr(x(:,i),x(:,i+1))
end

%归一化
[xn,PSx1] = mapminmax(x');
[yn,PSy] = mapminmax(y');

% 建立神经网络
net = feedforwardnet(5,'trainbr');
net = train(net, xn, yn); % 可能需要执行多次，若迭代次数为 1000，则需要执行下一次
```

```
31.
32. average_x=mean(x,1)
33. stanard_x=std(x,1)
34. data_pool_1=normrnd(average_x(1),stanard_x(1),100) %对于每个零件生成 100 个满足正态分布
    的数据并放入数据池
35. data_pool_2=normrnd(average_x(2),stanard_x(2),100)
36. data_pool_3=normrnd(average_x(3),stanard_x(3),100)
37. data_pool_4=normrnd(average_x(4),stanard_x(4),100)
38. data_pool_5=normrnd(average_x(5),stanard_x(5),100)
39. data_pool_6=normrnd(average_x(6),stanard_x(6),100)
40. data_pool_7=normrnd(average_x(7),stanard_x(7),100)
41. data_pool_8=normrnd(average_x(8),stanard_x(8),100)
42. data_pool_9=normrnd(average_x(9),stanard_x(9),100)
43. data_pool_10=normrnd(average_x(10),stanard_x(10),100)
44. T=1;
45. a=0.99; %模拟退火算法收敛系数
46. max=0;
47. r1=-3;  %设置初始值用于记录最终 x1~x10 选择
48. r2=-3;
49. r3=-3;
50. r4=-3;
51. r5=-3;
52. r6=-3;
53. r7=-3;
54. r8=-3;
55. r9=-3;
56. r10=-3;
57.
58. x1 = B1(1,:);%用于记录零件 1~10 的售价
59. x2 = B1(2,:);
60. x3 = B1(3,:);
61. x4 = B1(4,:);
62. x5 = B1(5,:);
63. x6 = B1(6,:);
64. x7 = B1(7,:);
65. x8 = B1(8,:);
66. x9 = B1(9,:);
67. x10 = B1(10,:);
68.
69. current_profit=0; %上次迭代的预期收益
70. flag=0; %是否停止退火
71. spmd
72.     for x_1=-2:2
73.         for x_2=-2:2
```

```
            for x_3=-2:2
                for x_4=-2:2
                    for x_5=-2:2
                        for x_6=-2:2
                            for x_7=-2:2
                                for x_8=-2:2
                                    for x_9=-2:2
                                        for x_10=-2:2
                                            if(T<0.5)
                                                flag=1;
                                            end
                                            cost=x1(abs(x_1)+1)+x2(abs(x_2)+1)+x3(abs(x_3)+1)+x4(abs(x_4)+1)+x5(abs(x_5)+1)+x6(abs(x_6)+1)+x7(abs(x_7)+1)+x8(abs(x_8)+1)+x9(abs(x_9)+1)+x10(abs(x_10)+1);
                                            X(1)=average_x(1)*(1+0.05*x_1); %设置x1~x10的初始值并遍历数据池
                                            X(2)=average_x(2)*(1+0.05*x_2);
                                            X(3)=average_x(3)*(1+0.05*x_3);
                                            X(4)=average_x(4)*(1+0.05*x_4);
                                            X(5)=average_x(5)*(1+0.05*x_5);
                                            X(6)=average_x(6)*(1+0.05*x_6);
                                            X(7)=average_x(7)*(1+0.05*x_7);
                                            X(8)=average_x(8)*(1+0.05*x_8);
                                            X(9)=average_x(9)*(1+0.05*x_9);
                                            X(10)=average_x(10)*(1+0.05*x_10);
                                            XN=mapminmax('apply',X',PSx1);
                                            fitnumber=0; %记录蒙特卡洛算法满足条件数
                                            y_nowall=mapminmax('reverse',sim(net,XN),PSy); %所有利润和
                                            for i=1:100
                                                if (data_pool_1(i)<average_x(1)*(1+0.05*(x_1+1)))&&(average_x(1)*(1+0.05*x_1)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
                                                    X(1)=data_pool_1(i);
                                                    y_nowall=y_nowall+mapminmax('reverse',sim(net,XN),PSy);
                                                end
                                            end
                                            for i=1:100
                                                if (data_pool_2(i)<average_x(2)*(1+0.05*(x_2+1)))&&(average_x(2)*(1+0.05*x_2)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
                                                    X(2)=data_pool_1(i);
```

```
                                                    y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                                end
                                            end
                                            for i=1:100
                                                if (data_pool_3(i)<average_x(3)*(1+0
    .05*(x_3+1)))&&(average_x(3)*(1+0.05*x_3)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
                                                    X(3)=data_pool_1(i);
                                                    y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                                end
                                            end
                                            for i=1:100
                                                if (data_pool_4(i)<average_x(4)*(1+0
    .05*(x_4+1)))&&(average_x(4)*(1+0.05*x_4)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
                                                    X(4)=data_pool_1(i);
                                                    y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                                end
                                            end
                                            for i=1:100
                                                if (data_pool_5(i)<average_x(5)*(1+0
    .05*(x_5+1)))&&(average_x(5)*(1+0.05*x_5)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
                                                    X(5)=data_pool_1(i);
                                                    y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                                end
                                            end
                                            for i=1:100
                                                if (data_pool_6(i)<average_x(6)*(1+0
    .05*(x_6+1)))&&(average_x(6)*(1+0.05*x_6)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
                                                    X(6)=data_pool_1(i);
                                                    y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                                end
                                            end
                                            for i=1:100
                                                if (data_pool_7(i)<average_x(7)*(1+0
    .05*(x_7+1)))&&(average_x(7)*(1+0.05*x_7)<data_pool_1(i))
                                                    fitnumber=fitnumber+1;
```

```
                                                X(7)=data_pool_1(i);
                                                y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                            end
                                        end
                                        for i=1:100
                                            if (data_pool_8(i)<average_x(8)*(1+0
    .05*(x_8+1)))&&(average_x(8)*(1+0.05*x_8)<data_pool_1(i))
                                                fitnumber=fitnumber+1;
                                                X(8)=data_pool_1(i);
                                                y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                            end
                                        end
                                        for i=1:100
                                            if (data_pool_9(i)<average_x(9)*(1+0
    .05*(x_9+1)))&&(average_x(9)*(1+0.05*x_9)<data_pool_1(i))
                                                fitnumber=fitnumber+1;
                                                X(9)=data_pool_1(i);
                                                y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                            end
                                        end
                                        for i=1:100
                                            if (data_pool_10(i)<average_x(10)*(1
    +0.05*(x_10+1)))&&(average_x(10)*(1+0.05*x_10)<data_pool_1(i))
                                                fitnumber=fitnumber+1;
                                                X(10)=data_pool_1(i);
                                                y_nowall=y_nowall+mapminmax('rev
    erse',sim(net,XN),PSy);
                                            end
                                        end
                                        y_now=y_nowall/fitnumber; %期望利润
                                        if (182<y_now)&&(y_now<186) %成品为 A 类
                                            profit=1500-cost;
                                        elseif ((178<y_now)&&(y_now<182))||((186
    <y_now)&&(y_now<190)) %成品为 B 类
                                            profit=1200-cost;
                                        elseif ((165<y_now)&&(y_now<178))||((190
    <y_now)&&(y_now<197)) %成品为 C 类
                                            profit=800-cost;
                                        else
                                            profit=600-cost; %成品为 D 类
                                        end
```

```
                                    if profit<current_profit
                                        T=a*T;
                                    end
                                    if profit>=max
                                        T=1;
                                        max=profit;
                                        disp("max="+max)
                                        r1=x_1;
                                        r2=x_2;
                                        r3=x_3;
                                        r4=x_4;
                                        r5=x_5;
                                        r6=x_6;
                                        r7=x_7;
                                        r8=x_8;
                                        r9=x_9;
                                        r10=x_10; %记录当前最大利润时零件取值
                                        %disp(XN)
                                        disp("gradex1~x10 ")
                                        disp([x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_10]')
                                        if T<0.5
                                            flag=1; %判断是否满足退火条件
                                        end
                                    end
                                    if profit>current_profit
                                        current_profit=profit; %记录上一次利润
                                    end
                                end
                                if flag==1
                                    break;
                                end
                            end
                            if flag==1
                                break;
                            end
                        end
                        if flag==1
                            break;
                        end
                    end
                    if flag==1
```

```
                                break;
                            end
                        end
                        if flag==1
                            break;
                        end
                    end
                    if flag==1
                        break;
                    end
                end
                if flag==1
                    break;
                end
            end
            if flag==1
                break;
            end
        end
        if flag==1
            break;
        end
    end
end
fprintf("10 end"+"\n")
delete(gcp)
```

模拟退火算法：

```
B1 = xlsread('2019B 附件.xlsx','附件 1');
x1 = B1(1,:);
x2 = B1(2,:);
x3 = B1(3,:);
x4 = B1(4,:);
x5 = B1(5,:);
x6 = B1(6,:);
x7 = B1(7,:);
x8 = B1(8,:);
x9 = B1(9,:);
x10 = B1(10,:);
average_x=mean(x,1);
T=1;
a=0.9995;
max=0;
```

```
r1=-3; %设置初始值用于记录最终 x1~x10 选择
r2=-3;
r3=-3;
r4=-3;
r5=-3;
r6=-3;
r7=-3;
r8=-3;
r9=-3;
r10=-3;
current_profit=0;
flag=0;
for x_1=-2:2
    for x_2=-2:2
        for x_3=-2:2
            for x_4=-2:2
                for x_5=-2:2
                    for x_6=-2:2
                        for x_7=-2:2
                            for x_8=-2:2
                                for x_9=-2:2
                                    for x_10=-2:2
                                        if(T<0.5)
flag=1;
end
cost=x1(abs(x_1)+1)+x2(abs(x_2)+1)+x3(abs(x_3)+1)+x4(abs(x_4)+1)+x5(abs(x_5)+1)+x6(abs(x_6)+1)+x7(abs(x_7)+1)+x8(abs(x_8)+1)+x9(abs(x_9)+1)+x10(abs(x_10)+1);
X3(1)=average_x(1)*(1+0.05*x_1);
X3(2)=average_x(2)*(1+0.05*x_2);
X3(3)=average_x(3)*(1+0.05*x_3);
X3(4)=average_x(4)*(1+0.05*x_4);
X3(5)=average_x(5)*(1+0.05*x_5);
X3(6)=average_x(6)*(1+0.05*x_6);
X3(7)=average_x(7)*(1+0.05*x_7);
X3(8)=average_x(8)*(1+0.05*x_8);
X3(9)=average_x(9)*(1+0.05*x_9);
X3(10)=average_x(10)*(1+0.05*x_10);
XN3 = mapminmax('apply', X3', PSx1);
y_now=mapminmax('reverse',sim(net,XN3), PSy);
                                        if(182<y_now)&&(y_now<186)
profit=1500-cost;
elseif ((178<y_now)&&(y_now<182))||((186<y_now)&&(y_now<190))
profit=1200-cost;
elseif ((165<y_now)&&(y_now<178))||((190<y_now)&&(y_now<197))
```

```
profit=800-cost;
                                        else
profit=600-cost;
end
                                        ifprofit<=current_profit
T=a*T;
end
                                        ifprofit>=max
T=1;
max=profit;
disp("max="+max)
r1=x_1;
r2=x_2;
r3=x_3;
r4=x_4;
r5=x_5;
r6=x_6;
r7=x_7;
r8=x_8;
r9=x_9;
r10=x_10;
%disp([x__1,x__2,x__3,x__4,x__5,x__6,x__7,x__8,x__9,x__10]')
%disp("gradex1~x10 ")
disp([x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_10])
                                            ifT<0.5
flag=1;
end
end
current_profit=profit;
end
                                    ifflag==1
                                        break;
end
end
                                ifflag==1
                                    break;
end
end
                            ifflag==1
                                break;
end
end
                        ifflag==1
                            break;
```

```
end
end
                    ifflag==1
                        break;
end
end
                ifflag==1
                    break;
end
end
            ifflag==1
                break;
end
end
        ifflag==1
            break;
end
end
    ifflag==1
        break;
end
end
fprintf("10 end"+"\n")
```