所属类别	2022 年“华数杯”全国大学生数学建模竞赛	参赛编号
本科		CM2204985

插层熔喷非织造材料的性能控制研究

摘要

近年来，熔喷非织造材料广泛用于口罩的生产制造，为了使产品性能更好，科学家对熔喷非织造材料进行插层处理。为此，我们拟通过对接收距离、热空气速度等理化因子的研究，预测在何种参数时，产品性能最佳。从而为插层熔喷非织造材料产品性能调控机制的建立提供一定的理论基础。

针对问题一，我们对厚度、孔隙率、回弹性率、过滤阻力、透气性、过滤效率等理化因子与插层率之间的关系进行分析，首先将插层前后六个理化因子分别与插层率绘制折线图，建立相关性分析模型，观察大致的相关关系，通过斯皮尔曼系数的分析对数据进行量化，对比数据变化率，求解得到插层率对这些变化的影响情况。

针对问题二，需要研究工艺参数和结构变量的关系，并据此预测结构变量数据。我们利用 bp 神经网络模型对结果进行预测，bp 神经网络具有理论上逼近任意非线性函数的能力，是一种按误差逆传播算法训练的多层前馈网络，能学习和贮存大量的输入-输出模式映射关系，并通过误差修正权重。首先我们输入原始数据，并设置训练数据和预测数据，随后构建 bp 神经网络，采用梯度下降法，利用 bp 神经网络进行预测，将预测结果反归一化并进行误差计算，进行误差分析和误差逆传播运算，得到更加精确的线性或非线性关系，获得其他参数对应的结构变量的预测值。

针对问题三，我们先选择典型相关性分析研究结构变量和产品性能之间的关系，反映两组变量之间相关性的强度。接下来为了研究结构变量之间、产品性能之间的关系，由于变量既不呈正态分布也没有线性关系，我们选择利用斯皮尔曼系数进行分析，求出同组变量两两之间的相关性。为了找到过滤效率最高的工艺参数，我们首先利用 matlab 中的 cftool 工具箱对数据进行拟合，得到过滤效率关于接收距离、热空气速度的多项式函数，然后利用粒子群优化算法求出最佳的工艺参数，即当接受距离为 18cm，热风速度为 1250r/min 时过滤效率达到最高。

针对问题四，我们先研究过滤阻力、过滤效率、厚度、压缩回弹性和工艺参数之间的关系。这里考虑到问题中的约束条件，以过滤阻力和过滤效率两个变量为目标函数，建立多目标模型，通过附件中的数据，利用 cftool 拟合工具箱求得各变量之间的关系式，利用 NSGA3 算法建立模型并进行优化，得到最佳工艺参数使得过滤效率尽可能高且过滤阻力尽可能低，即接受距离为 18cm，热空气速度为 1089r/min。

最后，我们基于优缺点对模型进行评价并加以改进推广。

关键词： 斯皮尔曼相关系数　bp 神经网络 粒子群优化算法 NSGA3 优化算法

一、问题重述

1.1 问题背景

熔喷非织造材料是目前广泛应用的一种空气过滤材料，其纤维细、材料比表面积大、蓬松性好、孔隙多，能够在过滤中起高效、低阻的过滤效果，对粉尘、细菌、固体颗粒物等物质的拦截效果好，是一种性能优越的过滤材料，其工艺简单且产量高，被广泛应用于口罩的制造中。但是，单一的熔喷非织造材料全部由细纤维构成，虽然过滤效率高，但是其密度较大、孔隙率较低，因而过滤阻力较大，纳污容量小。加入适量的粗纤维，使粗细纤维合理配置，能够有效改善熔喷非织造材料的过滤效果。因此，科学家们创造出插层熔喷法，制备出“Z 型”结构的插层熔喷非织造材料。

插层熔喷非织造材料制备的工艺参数较多，参数之间存在交互影响，加上插层气流后更为复杂，呈现出工艺参数决定结构变量，结构变量决定最终产品性能的特性。目前有关插层熔喷非织造材料的参数研究比较少，为了为产品性能调控机制建立一定的理论基础，寻找工艺参数、结构变量与产品性能的关系模型是一个非常重要的课题。

1.2 问题提出

基于相关文献和其他资料，本文将解决如下问题：

1.通过附件的数据，研究插层后结构变量、产品性能的变化规律和插层率对结构变量、产品性能的变化的影响。

2.研究工艺参数与结构变量之间的关系模型，并预测题中给定的工艺参数对应的结构变量数据。

3.研究结构变量与产品性能间的关系以及结构变量之间、产品性能之间的关系。由此建立工艺参数与产品性能的关系模型，并寻找过滤效率最高时产品的工艺参数。

4. 在满足接收距离不大于 100cm，热空气速度不大于 2000 r/min，厚度不超过 3mm， 压缩回弹性率不低于 85%的条件下，寻找使过滤效率尽量高且过滤阻力尽量小的最佳工艺参数。

二、问题分析

2.1 问题一的分析

分析插层后结构变量、产品性能的变化规律，首先明确了结构变量、产品性能两个明确指标，用 Excel 表格将附件中的数据（厚度、孔隙率、回弹性率、过滤阻力、透气性、过滤效率）制成折线图，观察大致的相关关系。然后对比数值变化值，进行统计分析，根据具体数据描述规律对指标的影响。 利用斯皮尔曼系数对其中的相关关系进行量化，得出更为可靠的结果。

2.2 问题二的分析

研究工艺参数和结构变量的关系并且预测结构变量数据，需要利用好所给的接受距离和热风速度数据，我们通过 matlab 实现 bp 神经网络的运用。首先我们将附件中数据

输入 matlab 软件中，设置训练数据和预测数据，通过构建 bp 神经网络，对工艺参数和结构变量之间的关系进行反误差分析，从已有数据中自动地归纳规则，获得这些数据的内在规律，得到相应的权值及阈值，最后通过表 1 中相关数据的输入，得到具有相当精度的预测量。

2.3 问题三的分析

可以将问题分成三个问题，即结构变量与产品性能的关系，结构变量之间、产品性能之间的关系，和过滤效率最高的问题。第一个小问题选择典型相关性分析的方法，分析两组变量之间的相关性；第二个小问题选择斯皮尔曼相关系数进行相关性的分析；第三个小问题首先用 matlab 对工艺参数和过滤效率进行曲面的拟合，得到拟合后的表达式，再选择粒子群优化算法处理表达式，求解出最优的工艺参数。

2.4 问题四的分析

研究过滤效率尽量高的同时过滤阻力尽可能小的问题，就是多目标优化问题，以过滤阻力尽可能小、过滤效率尽可能高为目标函数，利用 NSGA3 算法建立模型并进行优化。从而在约束条件下，求得过滤阻力小且过滤效率高的最优工艺参数。

三、模型假设

1. 假设所获取的数据全部真实有效
2. 假设所有数据均准确无误，没有统计过程中的错误和丢失
3. 在问题二的模型中，假设结构变量只与工艺参数有关，与其他变量无关。
4. 在问题三和问题四寻找最佳工艺参数的模型中，由于题目中给出工艺参数决定结构变量，结构变量决定产品性能的结论，在这里我们可以假设产品性能也是间接由工艺参数给出，与其他变量无关。

四、符号说明

符号	说明
r_s	spearman 相关系数
R^2	拟合优度
SSE	残差平方和
SSR	回归平方和
SST	总离差平方和
MAE	平均绝对误差
MSE	均方误差
RMSE	均方根误差
$x = (x_1, x_2, ..., x_n)$	输入向量
$hi = (hi_1, hi_2, ..., hi_p)$	隐含层输入向量
$ho = (ho_1, ho_2, ..., ho_p)$	隐含层输出向量
$yi = (yi_1, yi_2, ..., yi_q)$	输出层输入向量
$yo = (yo_1, yo_2, ..., yo_q)$	输出层输出向量
$d_o = (d_1, d_2, ..., d_q)$	期望输出向量

五、模型的建立与求解

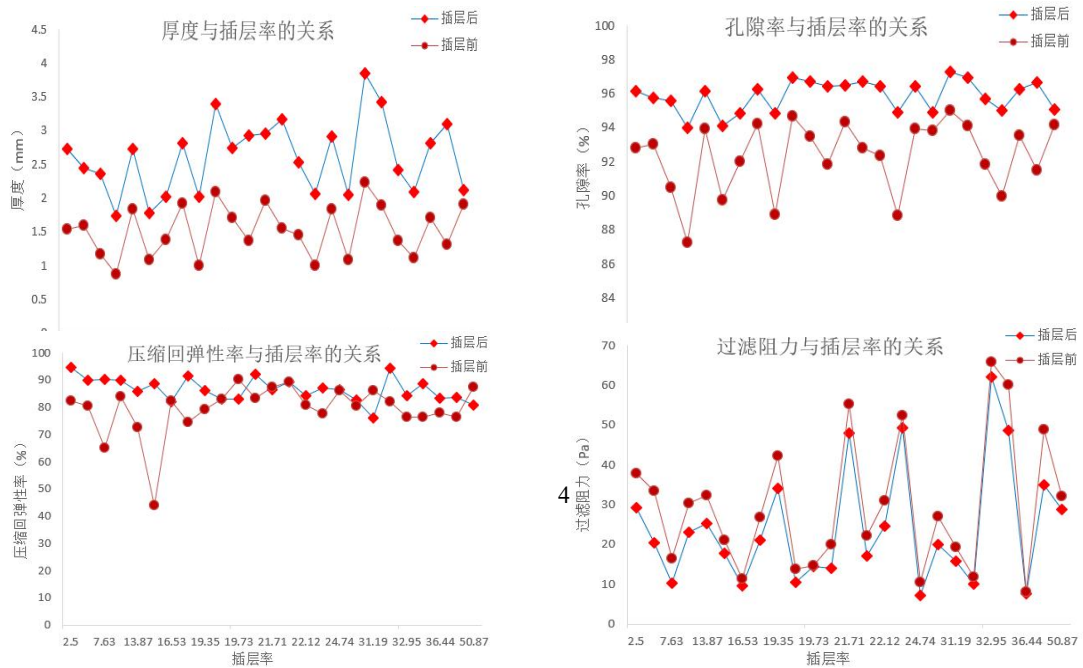
5.1 问题一的建模与求解

5.1.1 分析对象和指标的选取

问题一要求我们分析插层后结构变量、产品性能的变化规律，并分析插层率对这些变化的影响。经查阅资料可知，结构变量主要包括厚度、孔隙率、压缩回弹性，产品性能主要包括过滤阻力、过滤效率、透气性，由于普通熔喷非织造材料因为压缩回弹性差导致性能受影响，因此科学家制备出插层熔喷非织造材料。我们考虑分析未插层材料和插层材料结构变量和产品性能数据的变化，同时选取 data1 中的六个表示结构变量和产品性能的指标：厚度、孔隙率、压缩回弹性、过滤阻力、过滤效率、透气性进行分析。

5.1.2 折线图分析

我们用 EXCEL 分别对厚度、孔隙率、回弹性率、过滤阻力、过滤效率、透气性的变化用折线图进行表示，结果如下图。



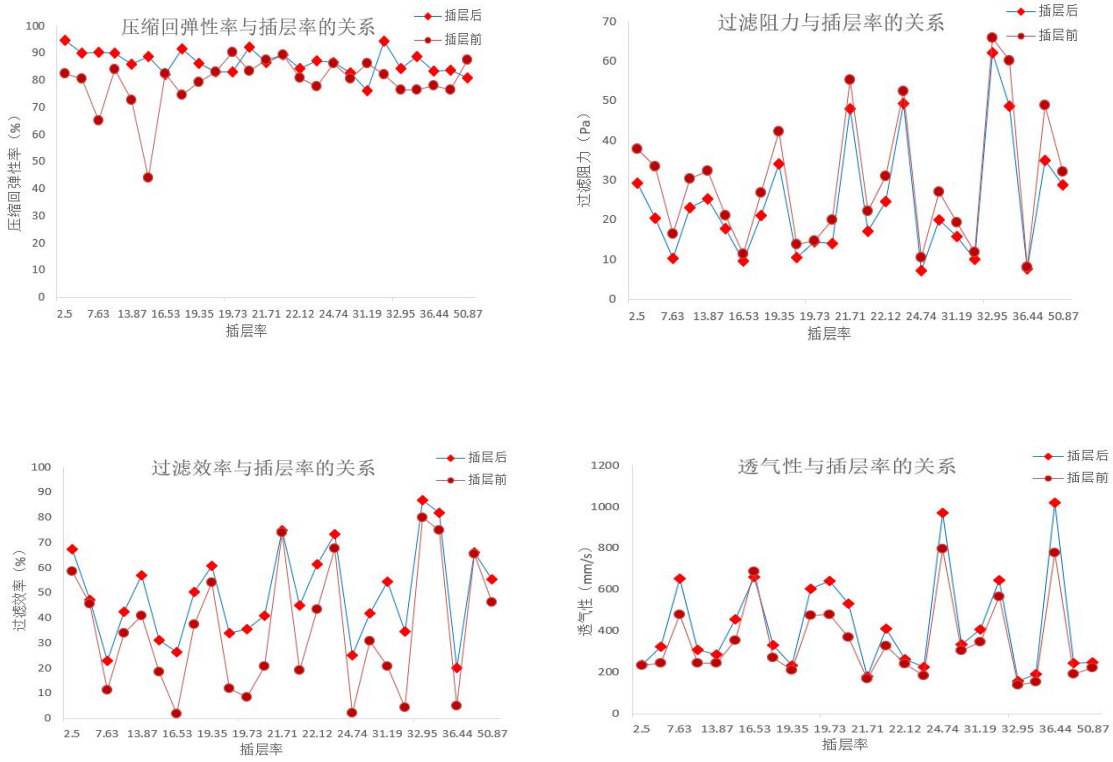


图 1　厚度等六个参数与插层率的关系

观察上图可以发现，与未插层材料相比，插层材料的厚度、孔隙率、压缩回弹性、过滤效率、透气性指标明显增加，而过滤阻力指标明显下降。

5.1.3 描述性分析

上面通过观察折线图我们可以大致得出变化规律，下面我们用 SPSS 对未插层材料和插层材料的厚度、孔隙率、压缩回弹性、过滤效率、过滤阻力、透气性进行描述性统计，定量计算出它们的最小值，最大值和平均值，结果如下表所示。

表 1 描述性统计结果

指标名称	样本个数	最小值统计	最大值统计	平均值
厚度（插层前）	25	1.74	3.85	2.61
厚度（插层后）	25	0.87	2.24	1.52
孔隙率（插层前）	25	94.00	97.30	95.86
孔隙率（插层后）	25	87.23	95.03	92.35
压缩回弹性率（插层前）	25	75.97	94.59	86.62
压缩回弹性率（插层后）	25	43.91	90.36	79.44
过滤阻力（插层前）	25	7.20	62.03	24.16
过滤阻力（插层后）	25	8.13	65.77	29.78
过滤效率（插层前）	25	19.97	86.97	49.39
过滤效率（插层后）	25	1.73	80.03	34.99
透气性 （插层前）	25	156.33	1019.67	422.13
透气性（插层后）	25	137.17	795.57	347.60
有效个案数（成列）	25			

通过观察上表平均值的变化，我们进一步明确了变化规律，即相比于未插层材料，插层材料的厚度、孔隙率、压缩回弹性、过滤效率、透气性增加，过滤阻力减小。

5.1.4 相关性分析

由上面折线图可大致地看出插层后数据变化的幅度。下面我们考虑用 spearman 相关系数进一步探讨插层率与各项指标的变化之间的关系，通过定量分析得到更可靠的结果。

5.1.4.1 数据处理

为了具体说明插层后数据变化的幅度，下面我们定义增长率的公式。

$$\text{厚度增长率}=\frac{\text{插层后厚度}-\text{插层前厚度}}{\text{插层前厚度}}$$

同理，我们可定义出孔隙率、压缩回弹性、过滤效率、透气性、过滤阻力的增长率。因此按照上述公式对 data1 中数据进行计算，可得到插层率对应的各项指标增长率，结果如下表所示。

表 2 各项指标增长率与插层率之间的关系

插层率（%）	厚度增长率	孔隙率增长率	压缩回弹性率增长率	过滤阻力增长率	过滤效率增长率	透气性增长率
36.44	0.638	0.030	0.069	-0.073	3.020	0.312
24.74	0.590	0.026	0.005	-0.312	11.916	0.218
31.45	0.812	0.030	0.149	-0.146	7.047	0.139
19.37	0.623	0.024	-0.002	-0.237	1.881	0.271
31.19	0.720	0.024	-0.117	-0.184	1.624	0.169
16.53	0.450	0.031	-0.002	-0.169	14.234	-0.037
19.73	0.613	0.034	-0.081	-0.007	3.184	0.346
21.72	1.042	0.042	-0.002	-0.229	1.337	0.252
18.62	0.467	0.022	0.230	-0.212	0.349	0.227
13.87	0.489	0.024	0.178	-0.221	0.395	0.175
7.63	1.021	0.056	0.387	-0.377	1.021	0.369
19.84	1.150	0.050	0.108	-0.293	0.977	0.444
4.84	0.528	0.029	0.119	-0.387	0.034	0.330
22.12	0.741	0.044	0.041	-0.203	0.413	0.096
2.50	0.769	0.037	0.149	-0.225	0.150	0.005
15.69	0.636	0.049	1.021	-0.154	0.692	0.292
29.79	0.898	0.012	0.025	-0.261	0.359	0.107
50.87	0.110	0.009	-0.073	-0.103	0.201	0.135
43.72	1.370	0.056	0.095	-0.287	0.011	0.260
21.71	0.503	0.022	-0.012	-0.132	0.014	0.067
11.32	1.000	0.078	0.074	-0.239	0.245	0.261
19.35	1.020	0.067	0.089	-0.191	0.127	0.108
24.02	1.070	0.069	0.124	-0.057	0.083	0.238
35.88	0.878	0.056	0.163	-0.193	0.092	0.239
32.95	0.777	0.042	0.105	-0.057	0.087	0.140

5.1.4.2 spearman 模型

Spearman 相关系数是衡量时间序列变化趋势在统计上有无显著性的常用方法，其原理是将两因子的样本值从小到大按顺序排列位次，以各因子样本值的位次代替实际数据加以计算。具体计算公式如下：

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

式中r_s代表 spearman 相关系数，n 为样本个数，若X_i与Y_i分别表示第 i 个样本的两组属性值，则d_i表示X_i与Y_i之间的等级差。当$|r_s|$<0.1 时,称 X,Y 具有无相关性；当 0.1<$|r_s|$<0.3 时，称 X,Y 具有弱相关性；当 0.3<$|r_s|$<0.5 时，称 X,Y 具有中相关性；当 0.5<$|r_s|$<1 时，称 X,Y 具有强相关性。用 SPSS 软件计算得到的相关系数和显著性 p 值如下表所示：

表 3 插层率与各项参数的相关系数及显著性 p 值表

	插层率	厚度	孔隙率	过滤阻力	压缩回弹性率	过滤效率	透气性
插层率	1.000(0.000)	0.172(0.421)	-0.127(0.554)	0.426(0.038)	-0.344(0.099)	-0.038(0.859)	-0.130(0.546)

由上表可以看出，插层率与过滤阻力的增长率在 0.05 水平上存在显著性的统计关系，插层率与压缩回弹性的增长率在 0.1 水平上存在显著性的统计关系，插层率与其他增长率不存在显著性的统计关系。因此，插层率与过滤阻力增长率、压缩回弹性增长率具有相关性，和其他增长率不存在相关性。

5.1.5 问题一的结果分析

综合前面的折线图、描述性统计、spearman 相关系数，我们可以得到结论：插层后厚度、孔隙率、压缩回弹性、过滤效率、透气性增加，过滤阻力减小；插层率与过滤阻力、压缩回弹性的变化有相关性，插层率越高，过滤阻力减小得越多，压缩回弹性增加得越多；插层率与厚度、孔隙率、过滤效率、透气性的变化几乎没有相关性。

5.2 问题二的建模与求解

5.2.1 分析对象和指标的选取

问题二要求我们研究工艺参数与结构变量的关系，同时根据表 1 中所给的数据进行预测。我们通过 matlab 实现 bp 神经网络，通过训练，神经网络将自主探究工艺参数与结构变量之间的关系，进而可用于预测工作。神经网络通过对输入数据深度学习，将输入输出之间的关系存于网络中，然后输入需要预测的数据，神经网络将输出预测结果

由于 bp 神经网络有自组织，自学习的特性，我们可以直接选取原始的接收距离以及热风速度作为输入，同时对这二者的相关性不做要求，模型自动完成特征提取，不需要人工的特征过程。通过 bp 神经网络误差逆传播和误差修正权数的特性的实践，得出较为精确的工艺参数与结构变量的线性或非线性关系，预测出精确度高的结构变量数据。

5.2.2 BP 神经网络模型

5.2.2.1 神经网络的定义简介

神经网络是由多个神经元组成的广泛互连的神经网络，能够模拟生物神经系统真实世界及物体之间所做出的交互反应.人工神经网络处理信息是通过信息样本对神经网络的训练，使其具有人的大脑的记忆，辨识能力，完成名种信息处理功能．它不需要任何先验公式，就能从已有数据中自动地归纳规则，获得这些数据的内在规律，具有良好的自学习，自适应，联想记忆，并行处理和非线性形转换的能力，特别适合于因果关系复杂的非确定性推理，判断，识别和分类等问题．对于任意一组随机的，正态的数据，都可以利用人工神经网络算法进行统计分析，做出拟合和预测.基于误差反向传播(Back propagation)算法的多层前馈网络(Multiple-layer feedforward network，简记为 BP 网络)，是目前应用最成功和广泛的人工神经网络.

5.2.2.2 BP 神经网络模型的基本原理

学习过程中由信号的正向传播与误差的逆向传播两个过程组成。正向传播时,模式作用于输入层，经隐层处理后，传入误差的逆向传播阶段，将输出误差按某种子形式，通过隐层向输入层逐层返回，并“分摊”给各层的所有单元，从而获得各层单元的参考误差或称误差信号，以作为修改各单元权值的依据.。权值不断修改的过程，也就是网络学习过程。此过程一直进行到网络输出的误差准逐渐减少到可接受的程度或达到设定的学习次数为止、BP 网络模型包括其输入输出模型，作用函数模型，误差计算模型和自学习模型。BP 网络由输入层，输出层以及一个或多个隐层节点互连而成的一种多层网，这种结构使多层前馈网络可在输入和输出间建立合适的线性或非线性关系，又不致使网

络输出限制在-1 和 1 之间。

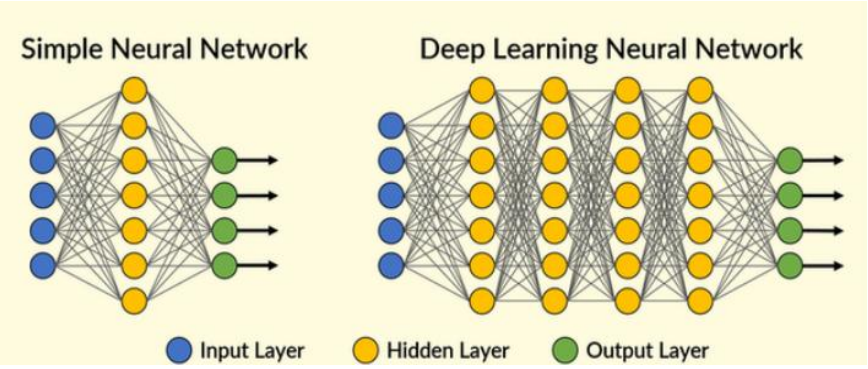


图 2 BP 网络模型示意图

5.2.2.3 BP 算法的过程

（1）首先计算各层神经元输入和输出

$$hi_h(k) = \sum_{i=0}^{n} w_{hi} x_i(k) \qquad h = 1, 2, \ldots\ldots, p$$

$$ho_h(k) = f(hi_h(k)) \qquad h = 1, 2, \ldots\ldots, p$$

$$yi_o(k) = \sum_{h=0}^{p} w_{oh} ho_h(k) \qquad o = 1, 2, \ldots\ldots, q$$

$$yo_o(k) = f\left(yi_o(k)\right) \qquad o = 1, 2, \ldots\ldots, q$$

（2） 利用网络期望输出和实际输出，计算误差函数对输出层的个神经元的偏导数

$$\frac{\partial e}{\partial yi_o} = \frac{\partial \left(\frac{1}{2}\sum_{o=1}^{q} (d_o(k) - yo_o(k))\right)^2}{\partial yi_o}$$

$$= -(d_o(k) - yo_o(k)) yo_o'(k)$$

$$= -(d_o(k) - yo_o(k)) f'(yi_o(k)) \triangleq -\delta_0(k)$$

$$\frac{\partial e}{\partial w_{oh}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{\partial w_{oh}}$$

$$\frac{\partial yi_o(k)}{\partial w_{oh}} = \frac{\partial (\sum_h^p w_{oh} ho_h(k))}{\partial w_{oh}} = ho_h(k)$$

（3）计算误差函数对隐含层个神经元的偏导数。

（4）修正连接权值

$$\Delta w_{hi}(k) = -\mu \frac{\partial e}{\partial w_{hi}} = \delta_h(k) x_i(k)$$

$$w_{hi}^{N+1} = w_{hi}^{N} + \mu\delta_h(k)x_i(k)$$

（5）计算全局误差

$$E=\frac{1}{2m}\sum_{k=1}^{m}\sum_{o=1}^{q}\left(d_o(k)-y_o(k)\right)^2$$

（6）判断网络误差是否满足要求，当误差达到预设精度或学习次数大于设定的最大次数，则算法结束。否则，选取下一个学习样本及对应的期望输出，返回下一轮学习。

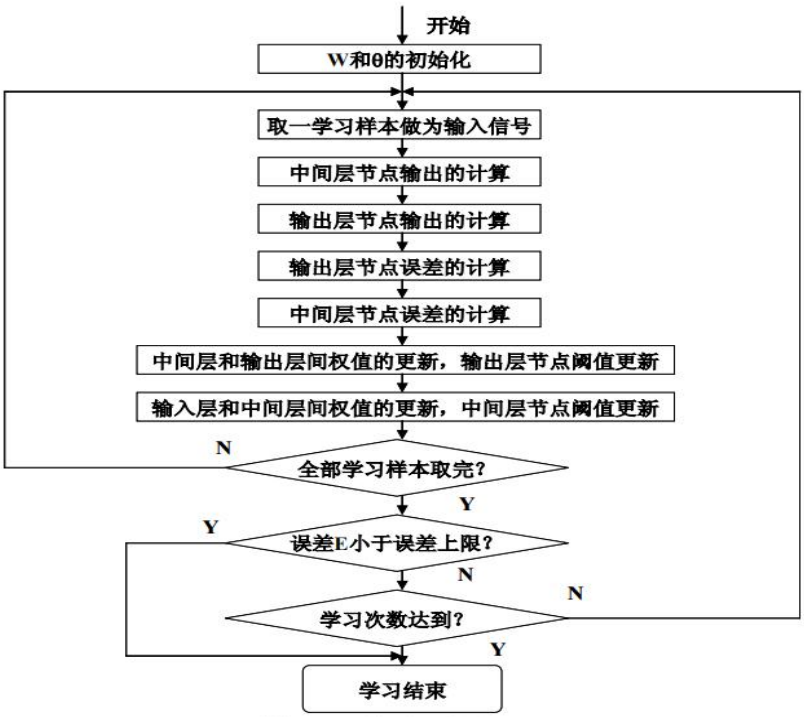


图 3 BP 算法流程图

5.2.2.4 原始数据选取

总数据共 75 组，数据数量较多，符合神经网络模型对于数据数量的要求。确定隐含层节点数的基本原则是：在满足精度的前提下，取尽可能紧凑的结构，即取尽可能少的隐含层节点数。在综合比对隐含层节点数在 4 到 15 时的平均绝对误差（MAE），均方误差（MSE）和均方根误差（RMSE），选取 7 个隐含层节点，选择 68 组数据对模型进行训练，然后使用另外 7 组数据的接收距离和热风速度，使用训练之后的模型进行预测，最终将预测值与实际值进行比对求取误差，来检测模型的训练效果。

表 4 不同隐含节点数的误差

隐含层节点	4	5	6	7	8	9
MAE	1.107	0.829	0.988	0.356	1.113	1.156
MSE	1.318	0.754	1.032	0.244	1.485	1.502
RMSE	1.148	0.868	1.016	0.493	1.218	1.225
隐含层节点	10	11	12	13	14	15
MAE	0.987	0.642	0.83	1.146	0.727	0.799

MSE	1.239	0.682	1.109	1.94	0.701	1.669
RMSE	1.113	0.826	1.053	1.392	0.837	1.291

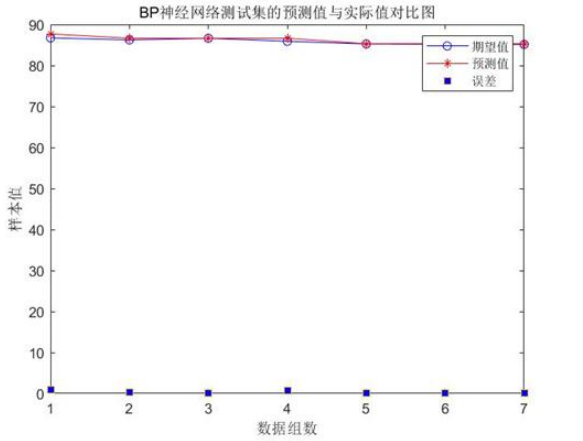


图 4 神经网络模型训练结果

5.2.3 问题二的结果分析

输入附件中所给的数据，利用 BP 逆误差传播和误差修正权数的特性，得出了较为精确的工艺参数与结构变量之间的关系：

利用所得关系，引入表 1 中 8 个接收距离和热风速度组合的数据，对厚度、孔隙率、压缩回弹性率进行预测，结果如下表所示：

表 5 厚度、孔隙率、压缩回弹性率预测结果表

接收距离(cm)	热风速度(r/min)	厚度 mm	孔隙率 （%）	压缩回弹性率%
38	850	2.742	96.146	86.540
33	950	2.651	96.727	87.826
28	1150	2.63	96.536	86.492
23	1250	2.561	96.042	86.186
38	1250	3.579	96.792	83.551
33	1150	3.132	96.408	86.731
28	950	2.312	95.958	88.424
23	850	1.943	94.144	87.376

5.3 问题三的建模与求解

5.3.1 分析对象和指标的选取

问题三要求我们研究结构变量与产品性能以及结构变量之间、产品性能之间的关系，同时求出过滤效率最高时的工艺参数。我们通过典型相关性分析探究结构变量与产品性能之间的关系，利用斯皮尔曼相关系数探究结构变量之间、产品性能之间的关系。最后利用 matlab 对工艺参数和过滤效率进行拟合，再利用粒子群优化算法求得最优的工艺参数。

5.3.2 基于典型相关性分析结构变量和产品性能之间的关系

5.3.2.1 典型相关性分析

典型相关分析基本思想和主成分分析非常相似。首先在每组变量中找出变量的线性组合，使得两组的线性组合之间具有最大的相关系数；然后选取和最初挑选的这对线性组合不相关的线性组合，使其配对，并选取相关系数最大的一对。如此继续下去，知道两组变量之间的相关性被提取完毕为止。被选出的线性组合配对称为典型变量，它们的相关系数称为典型相关系数。典型相关系数度量了这两组变量之间的强度。

一般情况下，假设

$$X^{(1)}=(X_1^{(1)},X_2^{(1)},\cdots,X_p^{(1)})、X^{(2)}=(X_1^{(2)},X_2^{(2)},\cdots,X_q^{(2)})$$

是两个相互关联的随机变量，分别在两组变量中选取若干有代表性的综合变量 Ui、Vi，使得每一个综合变量是原变量的线性组合，即

$$U_i=a_1^{(i)}X_1^{(1)}+a_2^{(i)}X_2^{(1)}+\cdots+a_P^{(i)}X_P^{(1)}\triangleq \mathbf{a}^{(i)'}\mathbf{X}^{(1)}$$

$$V_i=b_1^{(i)}X_1^{(2)}+b_2^{(i)}X_2^{(2)}+\cdots+b_q^{(i)}X_q^{(2)}\triangleq \mathbf{b}^{(i)'}\mathbf{X}^{(2)}$$

当然，综合变量的组数是不确定的，如果第一组就能代表原样本数据大部分的信息，那么一组就足够了。如果第一组反映的信息不够，我们就需要找第二组数据。

为了让所找到的第二组数据的信息更加有效，我们需要保证第二组数据和第一组数据不相关，即

$$cov(U_1,U_2)=cov(V_1,V_2)=0$$

5.3.2.2 结构变量和产品性能之间的关系分析结果

表 6 典型相关性

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.844	2.483	0.194	17.970	9.000	168.078	0.000
2	0.562	0.463	0.675	7.590	4.000	140.000	0.000
3	0.110	0.012	0.988	0.870	1.000	71.000	0.354

表 7 集合 1 标准化典型相关系数

变量	1	2	3
厚度	-1.092	1.642	-1.908
孔隙率	0.021	-1.216	2.143
压缩回弹性率	-0.156	1.265	0.135

表 8 集合 2 标准化典型相关系数

变量	1	2	3
过滤阻力	0.859	0.587	-0.361
过滤效率	0.640	-1.452	-0.390
透气性	0.437	-0.847	-1.361

表 9 集合 1 典型载荷

变量	1	2	3
厚度	-0.992	-0.116	-0.056
孔隙率	-0.909	-0.153	0.388
压缩回弹性率	0.413	0.794	0.447

表 10 集合 2 典型载荷

变量	1	2	3
过滤阻力	0.919	0.391	0.052
过滤效率	0.616	-0.564	0.549
透气性	-0.420	0.058	-0.906

表 11 已解释的方差比例

典型变量	集合 1* 自身	集合 1* 集合 2	集合 2* 自身	集合 2* 集合 1
1	0.660	0.470	0.467	0.333
2	0.222	0.070	0.158	0.050
3	0.118	0.001	0.375	0.005

结果分析：

根据典型相关性分析，可以知道在 99%置信水平下，结构变量和产品性能存在相关性，且前两对变量相关性显著，第三对变量相关性不显著。由此可以得到两对典型变量。

由典型载荷可以得到，结构变量的第一对典型变量与厚度、孔隙率负相关，而与压缩回弹性率正相关，其中与厚度的相关性最强；结构变量的第二对典型变量与厚度、孔隙率负相关，而与压缩回弹性率正相关。其中与压缩回弹性率的相关性最强。产品性能的第一对典型变量与过滤阻力、过滤效率正相关，与透气性负相关，其中与过滤阻力的相关性最强；产品性能的第二对典型变量与过滤阻力、透气性正相关，与过滤效率负相关，其中与过滤效率相关性最强。

由标准化典型相关系数可以得到，反映结构变量的典型变量主要由厚度决定，反映产品性能的第一对典型变量由过滤阻力决定，且过滤阻力和厚度呈负相关，即厚度越大阻碍物质颗粒通过熔喷非织造材料的能力越大，第二对典型变量主要由过滤效率决定，且过滤效率和厚度呈负相关，即厚度越大，物质颗粒通过熔喷非织造材料的比例越低，与客观事实相符。

5.3.3 基于 spearman 模型分析结构变量之间、产品性能之间的关系

5.3.3.1 spearman 相关系数

与问题一的模型不同，这里我们采用的是 spearman 相关系数第二种定义。spearman 相关系数被定义成等级变量之间的皮尔逊相关系数。对于样本容量为 n 的样本，n 个原

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}.$$

始数据被转换成等级数据，相关系数 ρ 为

spearman 相关系数表明 X(独立变量)和 Y(依赖变量)的相关方向。如果当 X 增加时，Y 趋向于增加，spearman 相关系数则为正。如果当 X 增加时，Y 趋向于减少，spearman 相关系数则为负。spearman 相关系数为零表明当 X 增加时 Y 没有任何趋向性。当 X 和 Y 越来越接近完全的单调相关时，spearman 相关系数会在绝对值上增加。当 X 和 Y 完全单调相关时，spearman 相关系数的绝对值为 1。完全的单调递增关系意味着任意两对数据 X_i，Y_i和X_j，Y_j，有X_i-X_j和Y_i-Y_J总是同号。完全的单调递减关系意味着任意两对数据X_i，Y_i和X_j，Y_j，有X_i-X_j和Y_i-Y_J总是异号。

5.3.3.2 基于 spearman 相关系数的分析结果

表12 结构变量相关性

			厚度	孔隙率	压缩回弹性率
斯皮尔曼 Rho	厚度	相关系数	1.000	0.921**	-0.473**
		显著性（双尾）	0.000	0.000	0.000
		N	75	75	75
	孔隙率	相关系数	0.921**	1.000	-0.355**
		显著性（双尾）	0.000	0.000	0.002
		N	75	75	75
	压缩回弹性率	相关系数	-0.473**	-0.355**	1.000
		显著性（双尾）	0.000	0.002	0.000
		N	75	75	75

**. 在 0.01 级别（双尾），相关性显著。

表13 产品性能相关性

			过滤阻力Pa	过滤效率（%）	透气性 mm/s
斯皮尔曼 Rho	过滤阻力Pa	相关系数	1.000	0.789**	-0.648**
		显著性（双尾）	0.000	0.000	0.000
		N	75	75	75
	过滤效率（%）	相关系数	0.789**	1.000	-0.778**
		显著性（双尾）	0.000	0.000	0.000
		N	75	75	75
	透气性 mm/s	相关系数	-0.648**	-0.778**	1.000
		显著性（双尾）	0.000	0.000	0.000
		N	75	75	75

**. 在 0.01 级别（双尾），相关性显著。

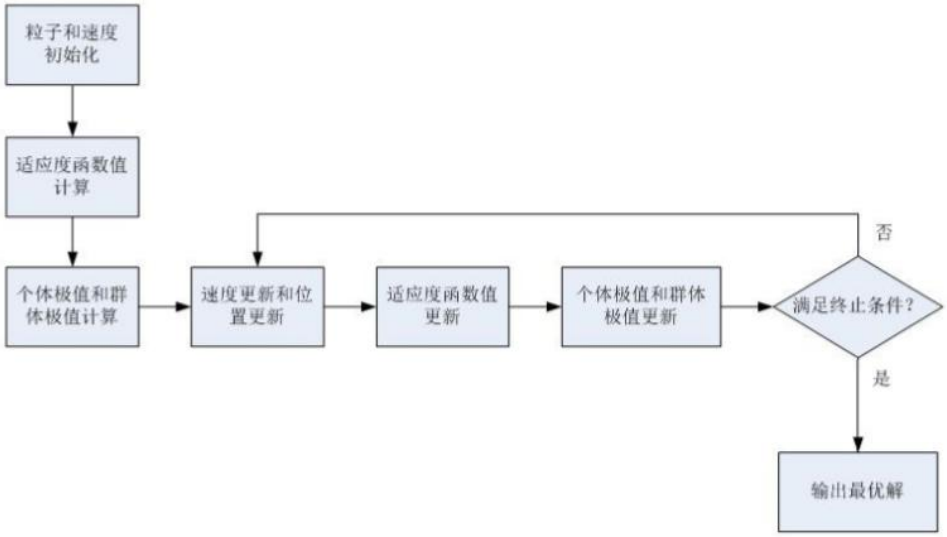
由结果可以得到，结构变量之间是相关的，其中厚度和孔隙率之间正相关，厚度和压缩回弹性率、孔隙率和压缩回弹性率之间负相关，其中厚度和孔隙率之间相关性最强。产品性能之间也是相关的，其中过滤阻力和过滤效率正相关，过滤阻力和透气性、过滤效率和透气性负相关，其中过滤阻力和过滤效率相关性最强。

5.3.4 基于粒子群优化算法求解最优工艺参数

5.3.4.1 粒子群优化算法

粒子群优化（PSO, particle swarm optimization）算法是计算智能领域，除了蚁群算法，鱼群算法之外的一种群体智能的优化算法，该算法最早由 Kennedy 和 Eberhart 在 1995 年提出的，该算法源自对鸟类捕食问题的研究。PSO 算法首先在可行解空间中初始化一群粒子，每个粒子都代表极值优化问题的一个潜在最优解，用位置、速度和适应度值三项指标表示该粒子特征。粒子在解空间中运动，通过跟踪个体极值 Pbest 和群体极值 Gbest 更新个体位置，个体极值 Pbest 是指个体所经历位置中计 算得到的适应度值最优位置，群体极值 Gbest 是指种群中的所有粒子搜索到的适应度最优位置。粒子每更新一次位置，就计算一次适应度值，并且通过比较新粒子的适应度值和个体极值、群体极值的适应度值更新个体 极值 Pbest 和群体极值 Gbest 位置。

在每一次迭代过程中，粒子通过个体极值和群体极值更新自身的速度和位置，更新



公式。如下：

图 5 粒子群优化算法流程图

5.3.4.2 拟合曲面及求解

建立过滤效率与接收距离、热空气速度之间关系的数学模型，考虑应用 matlab 中的 cftool 工具箱拟合曲线进行分析，我们可得到如下拟合函数：

$$f(x,y) = 5652 - 69.3 * x - 21.24 * y + 3.271 * x^2 + 0.0473 * x * y + 0.03178 * y^2 - 0.1418 * x^3 + 0.00623 * x^2 * y - 0.0002633 * x * y^2 - 1.857 * 10^{-5} * y^3 + 0.001584 * x^4 - 5.679 * 10^{-5} * x^3 * y - 2.689 * 10^{-7} * x^2 * y^2 + 9.601 * 10^{-8} * x * y^3 + 3.854 * 10^{-9} * y^4$$

其中$f(x,y)$表示过滤效率，x表示接收距离，y表示热空气速度

此拟合函数的拟合优度 R^2 为 0.9562，误差平方和 SSE 为 376.1319, 拟合精确度较高，可以作为描述过滤效率与接收距离、热空气速度的关系式。

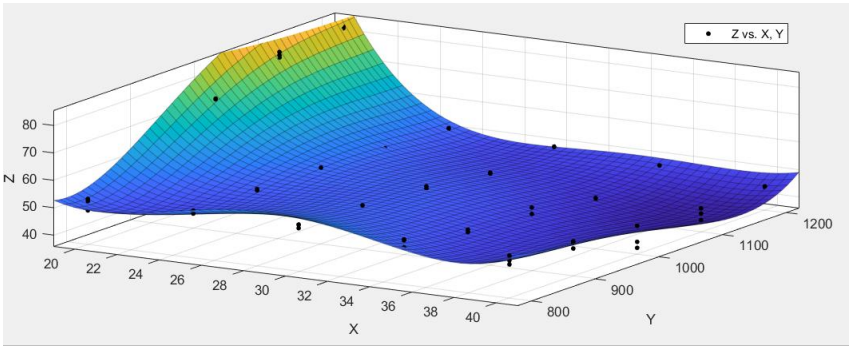


图 6 过滤效率拟合曲面

通过粒子群算法对该模型进行优化求解，求得结果如下：

表 14 过滤效率最高时的工艺参数

接收距离(cm)	热风速度(r/min)	过滤效率（%）
18	1250	88.166159

5.4 问题四的建模与求解

5.4.1 拟合曲面

由于上一问中我们已经建立过滤效率与工艺参数间的关系式，现在我们建立过滤阻力与接收距离、热空气速度之间关系的数学模型，考虑应用 matlab 中的 cftool 工具箱拟合曲线进行分析，我们可得到如下拟合函数：

$$f(x,y) = 29.66 - 4.721 * x - 3.464 * y - 3.381 * x^2 - 1.76 * x * y + 0.9475 * y^2 + 0.8683 * x^3 + 0.8449 * x^2 * y + 0.4588 * x * y^2 + 0.2721 * y^3 + 1.231 * x^4 + 0.6776 * x^3 * y - 0.4783 * x^2 * y^2 + 0.4543 * x * y^3 + 0.04412 * y^4$$

其中$f(x,y)$表示过滤阻力，x表示接收距离，y表示热空气速度

此拟合函数的拟合优度 R^2 为 0.7188，误差平方和 SSE 为 440.6838，拟合精确度较高，可以作为描述过滤阻力与接收距离、热空气速度的关系式。

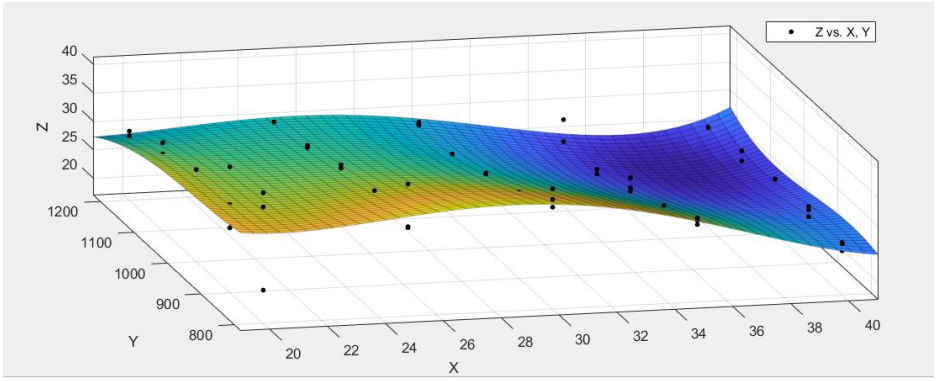


图 7 过滤阻力拟合曲面、

5.4.2 NSGA3 算法

5.4.2.1 NSGA3 算法基本原理

NSGA 相比于普通遗传算法,该算法在选择算子执行之前根据个体之间的支配关系进行了分层。其选择算子、交叉算子和变异算子与简单遗传算法没有区别。

而 NSGAII 和 NSGA 相比，它使用了精英策略，即将父代种群与其产生的子代种群组合，共同竞争产生下一代种群，有利于保持父代中的优良个体进入下一代，并通过对种群中所有个体的分层存放，使得最佳个体不会丢失，迅速提高种群水平；提出了拥挤度和拥挤度比较算子，代替了需要指定共享半径的适应度共享策略。

NSGAIII 和 NSGAII 具有类似的框架，二者区别主要在于选择机制的改变，NSGAII 主要靠拥挤度进行排序，其在高维目标空间显然作用不太明显;而 NSGAIII 对拥挤度排序进行了大刀阔斧的改编，通过引入广泛分布的参考点来维持种群的多样性。

初始的时候，随机生成父种群 Pt（N），再经过交叉和变异产生子种群（N），两者结合起来，用非支配的排序方法将它们划分等级（F1、F2、…Fl），从 F1 层开始选择个体，然后是 F2……，把它们放在 St 中，一直到 St 的大小为 N，作为下一次迭代的父种群 Pt+1。

在取到最后一层的时候，只选择其中的部分——K=N-|Pt+1|个。NSGAIII 使用的方法便是预先指定参考线，计算 St 中的个体与参考线之间的垂直距离。步骤如下：

1.对 St 中的值进行自适应归一化

2.定义超平面上连接参考点的参考线

3.计算 St 中的个体与参考线之间的垂直距离

4.每个个体都根据最小垂直距离与一个参考点相关联

5.计算每个参考点相关联的个体数量

6.根据第五步数量选择 K 个个体

5.4.2.2 NSGA3 算法流程

NSGA-III 首先定义一组参考点。然后随机生成含有 N 个个体的初始种群，其中 N 是种群大小。接下来，算法进行迭代直至终止条件满足。在第 t 代，算法在当前种群 Pt 的基础上，通过随机选择，模拟两点交叉和多项式变异 产生子代种群 Qt。Pt 和 Qt 的大小均为 N。因此，两个种群 Pt 和 Qt 合并会形成种群大小为 2N 的新的种群 Rt=Pt∪Qt。

为了从种群 Rt 中选择最好的 N 个解进入下一代，首先利用基于 Pareto 支配的非支配排序将 Rt 分为若干不同的非支配层（F1，F2 等等）。然后，算法构建一个新的种群 St，构建方法是从 F1 开始，逐次将各非支配层的解加入到 St，直至 St 的大小等于 N，或首次大于 N。假设最后可以接受的非支配层是 L 层，那么在 L+ 1 层以及之后的那些解就被丢弃掉了，且 St\ FL 中的解已经确定被选择作为 Pt+1 中的解。Pt+1 中余下的个体需要从 FL 中选取，选择的依据是要使种群在目标空间中具有理想的多样性。

NSGA 的思想首先是把所有解进行分非支配排序，这里目标值越小越好，甲三个目标值是（2，3，5），乙的三个目标是（4，3，5），丙的三个目标是（3，3，4），那么给甲和丙在 F1 层即等级为 1，F2 层是乙，等级为 2，以此类推。假如一共有 8 层非支配层（F1，F2，…，F8），你选到 F1+F2+F3+…+F7 时候已经有 90 个个体了，而 F8 层有 20 个个体，那么怎么在 F8 选这 10 个个体，即怎么在 F8 里的 20 个选十个，NSGA2 用的是基于拥挤距离的方法，而 NSGA3 用的是基于参考点的方法。（ St\ FL 里面，St 就是

这里的 F1 到 F7 的所有个体的总和，F8 就是 FL，St\ FL 这个符号意思就是已经被选择的 F1 到 F7 所有个体）

5.4.2.3 基于 NSGA3 算法求解

通过 NSGA3 算法对该模型进行优化求解，求得结果如下
：

表 15 过滤效率高且过滤阻力小时的工艺参数结果

接收距离(cm)	热风速度(r/min)	过滤效率（%）	过滤阻力Pa
18	1089	87.159	20.865

六、模型的评价与推广

6.1 BP 神经网络

6.1.1 BP 神经网络的优缺点

6.1.1.1 BP 神经网络的优点

1）非线性映射能力：BP 神经网络实质上实现了一个从输入到输出的映射功能，数学理论证明三层的神经网络就能够以任意精度逼近任何非线性连续函数。这使得其特别适合于求解内部机制复杂的问题，即 BP 神经网络具有较强的非线性映射能力。

2）自学习和自适应能力：BP 神经网络在训练时，能够通过学习自动提取输出、输出数据间的“合理规则”，并自适应的将学习内容记忆于网络的权值中。即 BP 神经网络具有高度自学习和自适应的能力。

3）泛化能力：所谓泛化能力是指在设计模式分类器时，即要考虑网络在保证对所需分类对象进行正确分类，还要关心网络在经过训练后，能否对未见过的模式或有噪声污染的模式，进行正确的分类。也即 BP 神经网络具有将学习成果应用于新知识的能力。

4）容错能力：BP 神经网络在其局部的或者部分的神经元受到破坏后对全局的训练结果不会造成很大的影响，也就是说即使系统在受到局部损伤时还是可以正常工作的。即 BP 神经网络具有一定的容错能力。

6.1.1.2 BP 神经网络的缺点

1）局部极小化问题：从数学角度看，传统的 BP 神经网络为一种局部搜索的优化方法，它要解决的是一个复杂非线性化问题，网络的权值是通过沿局部改善的方向逐渐进行调整的，这样会使算法陷入局部极值，权值收敛到局部极小点，从而导致网络训练失败。加上 BP 神经网络对初始网络权重非常敏感，以不同的权重初始化网络，其往往会收敛于不同的局部极小，这也是我们每次训练得到不同结果的根本原因。

2）BP 神经网络算法的收敛速度慢：由于 BP 神经网络算法本质上为梯度下降法，它所要优化的目标函数是非常复杂的，因此，必然会出现“锯齿形现象”，这使得 BP 算法低效；又由于优化的目标函数很复杂，它必然会在神经元输出接近 0 或 1 的情况下，出现一些平坦区，在这些区域内，权值误差改变很小，使训练过程几乎停顿；BP 神经网络模型中，为了使网络执行 BP 算法，不能使用传统的一维搜索法求每次迭代的步长，而必须把步长的更新规则预先赋予网络，这种方法也会引起算法低效。以上种种，导致了 BP 神经网络算法收敛速度慢的现象。

3）BP 神经网络结构选择不一：BP 神经网络结构的选择至今尚无一种统一而完整的理论指导，一般只能由经验选定。网络结构选择过大，训练中效率不高，可能出现过拟

合现象，造成网络性能低，容错性下降，若选择过小，则又会造成网络可能不收敛。而网络的结构直接影响网络的逼近能力及推广性质。因此，应用中如何选择合适的网络结构是一个重要的问题。

4）应用实例与网络规模的矛盾问题：BP 神经网络难以解决应用问题的实例规模和网络规模间的矛盾问题，其涉及到网络容量的可能性与可行性的关系问题，即学习复杂性问题。

5）BP 神经网络预测能力和训练能力的矛盾问题：预测能力也称泛化能力或者推广能力，而训练能力也称逼近能力或者学习能力。一般情况下，训练能力差时，预测能力也差，并且一定程度上，随着训练能力地提高，预测能力会得到提高。但这种趋势不是固定的，其有一个极限，当达到此极限时，随着训练能力的提高，预测能力反而会下降，也即出现所谓“过拟合”现象。出现该现象的原因是网络学习了过多的样本细节导致，学习出的模型已不能反映样本内含的规律，所以如何把握好学习的度，解决网络预测能力和训练能力间矛盾问题也是 BP 神经网络的重要研究内容。

6） BP 神经网络样本依赖性问题：网络模型的逼近和推广能力与学习样本的典型性密切相关，而从问题中选取典型样本实例组成训练集是一个很困难的问题。

6.1.2 BP 神经网络的改进

优化网络学习率、网络初始参数、网络学习结构的参数来提高训练速度。修正 BP 算法的误差函数和激励函数。网络初始参数的归一化处理。与遗传算法、模拟退火算法等其他算法结合。

6.1.3 BP 神经网络的推广应用

1)函数逼近：用输入向量和相应的输出向量训练一个网络逼近一个函数。

2)模式识别：用一个待定的输出向量将它与输入向量联系起来。

3)分类：把输入向量所定义的合适方式进行分类。

4)数据压缩：减少输出向量维数以便于传输或存储。

6.2 粒子群优化算法

6.2.1 粒子群优化算法的优缺点

6.2.1.1 粒子群优化算法的优点

PSO 同遗传算法类似，是一种基于迭代的优化算法。系统初始化为一组随机解，通过迭代搜寻最优值。同遗传算法比较，PSO 的优势在于简单容易实现，并且没有许多参数需要调整。

6.2.1.2 粒子群优化算法的缺点

在某些问题上性能并不是特别好。网络权重的编码而且遗传算子的选择有时比较麻烦。

6.2.2 粒子群优化算法的改进

经典的粒子群算法的寻优性能在很大程度上依赖于其惯性权重、学习因子参数的设置选择。由此，初始粒子的每一次迭代都进行了空间位置、速度的更新，忽略了不同代粒子间的差异性，应用了统一的固定参数 。

针对于这一问题在参数设置上：采用了线性递减的参数设置，以较大的惯性权重和学习因子为开局，有利于增加初始粒子群的全局迭代寻优能力；以较小的惯性权重和学习因子为结束，有利于加强最终粒子群跳出局部最好解。

（1） 通常对粒子群算法的参数设置进行改进
（2） 融合其他算法，形成混合算法

6.2.3 粒子群优化算法的推广应用

（1）函数优化
（2）神经网络训练
（3）用于随机优化问题的求解
（4）用于最优控制问题的求解

七、参考文献

[1]钱晨坚. Spearman 秩相关系数比较的假设检验及样本量估计新方法研究[D].南方医科大学,2021.DOI:10.27003/d.cnki.gojyu.2021.000765.

[2]冯茜,李擎,全威,裴轩墨.多目标粒子群优化算法研究综述[J].工程科学学报,2021,43(06):745-753.DOI:10.13374/j.issn2095-9389.2020.10.31.001.

[3]杨君岐,任瑞,阚立娜,任昊悦.基于 BP 神经网络模型的商业银行风险评估研究[J].会计之友,2021(05):113-119.

[4]韩玲,胡梦缘,马英博,郝栋连.医用非织造口罩材料及其新技术的研究现状[J].西安工程大学学报,2020,34(02):20-25.DOI:10.13338/j.issn.1674-649x.2020.02.003.

[5]邹志伟. 插层熔喷气流场模拟及生产工艺参数的优化[D].天津工业大学,2020.DOI:10.27357/d.cnki.gtgyu.2020.000926.

[6]杨盼,卢路,王继保,陈和春.基于主成分分析的 spearman 秩相关系数法在长江干流水质分析中的应用[J].环境工程,2019,37(08):76-80.DOI:10.13205/j.hjgc.201908014.

[7]李捷菲. 基于 BP 神经网络的 PID 控制系统研究与设计[D].吉林大学,2019.

[8]武辉. 插层熔喷气流场模拟及其过滤材料性能的研究[D].天津工业大学,2018.

[9]居凤霞. 粒子群优化算法的改进及应用[D].华南理工大学,2014.

附录

附录 1 问题二 bp 神经网络的构建及结构变量数据的预测（matlab）

```
%% 此程序为 matlab 编程实现的 BP 神经网络
% 清空环境变量
% clear
close all  %关闭所有图形窗口
clc


%%第一步 读取数据
X=xlsread('C 题数据.xlsx',3,'A2:B84');
Y=xlsread('C 题数据.xlsx',3,'E2:E84');
input=X;   %载入输入数据
output=Y;  %载入输出数据


%% 第二步 设置训练数据和预测数据
% 注意要将指标变为列向量
input_train = input(1:65,:)';
output_train =output(1:65,:)';
input_test = input(66:83,:)';
output_test =output(66:83,:)';
%节点个数
inputnum=2; % 输入层节点数量
hiddennum=18; % 隐含层节点数量
outputnum=1;  % 输出层节点数量
%% 第三本 训练样本数据归一化
[inputn,inputps]=mapminmax(input_train);%归一化到[-1,1]之间，inputps 用来作下一次同样的归一化
[outputn,outputps]=mapminmax(output_train);
%% 第四步 构建 BP 神经网络
net=newff(inputn,outputn,hiddennum,{'tansig','purelin'},'trainlm');% 建立模型，传递函数使用 purelin，采用梯度下降法训练


W1= net. iw{1, 1};                  %输入层到中间层的权值
B1 = net.b{1};                      %中间各层神经元阈值


W2 = net.lw{2,1};                   %中间层到输出层的权值
B2 = net. b{2};                     %输出层各神经元阈值
```

```
%% 第五步 网络参数配置（ 训练次数，学习速率，训练目标最小误差等）
net.trainParam.epochs=1000;         % 训练次数，这里设置为 1000 次
net.trainParam.lr=0.01;             % 学习速率，这里设置为 0.01
net.trainParam.goal=0.00001;        % 训练目标最小误差，这里设置为 0.00001


%% 第六步 BP 神经网络训练
net=train(net,inputn,outputn);%开始训练，其中 inputn,outputn 分别为输入输出样本


%% 第七步 测试样本归一化
inputn_test=mapminmax('apply',input_test,inputps); % 对样本数据进行归一化


%% 第八步 BP 神经网络预测
an=sim(net,inputn_test);                          %用训练好的模型进行仿真


%% 第九步 预测结果反归一化与误差计算
test_simu=mapminmax('reverse',an,outputps);        %把仿真得到的数据还原为原始的
数量级
error=test_simu-output_test;                       %预测值和真实值的误差
test_simu
%%第十步 真实值与预测值误差比较
figure('units','normalized','position',[0.119 0.2 0.38 0.5])
plot(output_test,'bo-')
hold on
plot(test_simu,'r*-')
hold on
plot(error,'square','MarkerFaceColor','b')
legend('期望值','预测值','误差')
xlabel('数据组数')
ylabel('样本值')
title('BP 神经网络测试集的预测值与实际值对比图')


[c,l]=size(output_test);
MAE1=sum(abs(error))/l;
MSE1=error*error'/l;
RMSE1=MSE1^(1/2);
disp(['-----------------------误差计算--------------------------'])
disp(['隐含层节点数为',num2str(hiddennum),'时的误差结果如下：'])
disp(['平均绝对误差 MAE 为：',num2str(MAE1)])
disp(['均方误差 MSE 为：       ',num2str(MSE1)])
disp(['均方根误差 RMSE 为：   ',num2str(RMSE1)])
```

附录 2 问题三 粒子群优化算法（matlab）

```
function main
clear all;
close all;
%（1）初始化粒子群算法参数
min=18;max=1250;%粒子位置范围
Vmax=25;Vmin=-25;%粒子运动速度范围
c1=1.3;c2=1.7;  %学习因子[0，4]
wmin=0.20;wmax=0.90;%惯性权重
G=400;          % 最大迭代次数
Size=100;        %初始化群体个体数目
for i=1:G
    w(i)=wmax-((wmax-wmin)/G)*i;  %随着优化进行，应降低自身权重
end

for i=1:Size
    for j=1:2
        x(i,j)=min+(max-min)*rand(1);     %随机初始化位置
        v(i,j)=Vmin +(Vmax-Vmin)*rand(1); %随机初始化速度
    end
end

%（2）计算各个粒子的适应度，并初始化 Pi、plocal 和最优个体 BestS
for i=1:Size
    p(i)=func(x(i,:));
    y(i,:)=x(i,:);

    if i==1
        plocal(i,:)=evaluate_localbest(x(Size,:),x(i,:),x(i+1,:));
    elseif i==Size
        plocal(i,:)=evaluate_localbest(x(i-1,:),x(i,:),x(1,:));
        else
        plocal(i,:)=evaluate_localbest(x(i-1,:),x(i,:),x(i+1,:));
    end
end

BestS=x(1,:);%初始化最优个体 BestS
for i=2:Size
    if func(x(i,:))>func(BestS)
        BestS=x(i,:);
    end
end

%（3）进入主循环
```

```
for kg=1:G
    for i=1:Size

     M=1;
     if M==1

v(i,:)=w(kg)*v(i,:)+c1*rand*(y(i,:)-x(i,:))+c2*rand*(plocal(i,:)-x(i,:));%局部寻优：加权，实现速度的更新
     elseif M==2

v(i,:)=w(kg)*v(i,:)+c1*rand*(y(i,:)-x(i,:))+c2*rand*(BestS-x(i,:));      %全局寻优：加权，实现速度的更新
     end
         for j=1:2   %检查速度是否越界
           if v(i,j)<Vmin
               v(i,j)=Vmin;
           elseif  x(i,j)>Vmax
               v(i,j)=Vmax;
           end
         end
       x(i,:)=x(i,:)+v(i,:)*1; %实现位置的更新
       for j=1:2   %检查位置是否越界
           if x(i,j)<min
               x(i,j)=min;
           elseif  x(i,j)>max
               x(i,j)=max;
           end
        end
%自适应变异,避免粒子群算法陷入局部最优
      if rand>0.60
          k=ceil(2*rand);
          x(i,k)=min+(max-min)*rand(1);
      end
%（4）判断和更新
      if i==1
          plocal(i,:)=evaluate_localbest(x(Size,:),x(i,:),x(i+1,:));
       elseif i==Size
          plocal(i,:)=evaluate_localbest(x(i-1,:),x(i,:),x(1,:));
       else
          plocal(i,:)=evaluate_localbest(x(i-1,:),x(i,:),x(i+1,:));
       end

      if func(x(i,:))>p(i) %判断当此时的位置是否为最优的情况，当不满足时继续更新
        p(i)=func(x(i,:));
```

```
        y(i,:)=x(i,:);
      end
      if p(i)>func(BestS)
          BestS=y(i,:);
      end
   end
Best_value(kg)=func(BestS);
end
figure(1);
kg=1:G;
plot(kg,-Best_value,'r','linewidth',2);
xlabel('generations');ylabel('Fitness function');


% display('Best Sample=');disp(BestS);
display('X=');disp(BestS(1));
display('Y=');disp(BestS(2));
% display('Biggest value=');disp(Best_value(G));
display('Z=');disp(Best_value(G));
for i=1:G
    if Best_value(i)==Best_value(end)
        disp('收敛到最优值需要的迭代步数');
        disp(i);
        break;
    end
end

end
```

```
function f = func(x)
  p00 = 5652  ;
  p10 =      -69.3 ;
  p01 =     -21.24  ;
  p20 =      3.271 ;
  p11 =     0.0473 ;
  p02 =    0.03178 ;
  p30 =    -0.1418 ;
  p21 =    0.00623  ;
  p12 =  -0.0002633  ;
  p03 =  -1.857e-05  ;
  p40 =   0.001584 ;
  p31 =  -5.679e-05 ;
```

```
  p22 =  -2.689e-07 ;
  p13 =   9.601e-08  ;
  p04 =   3.883e-09  ;


f= p00 + p10*x(1)+ p01*x(2) + p20*x(1)^2 + p11*x(1)*x(2) + p02*x(2)^2 + p30*x(1)^3
+ p21*x(1)^2*x(2) + p12*x(1)*x(2)^2 + p03*x(2)^3 + p40*x(1)^4 + p31*x(1)^3*x(2)
+ p22*x(1)^2*x(2)^2 + p13*x(1)*x(2)^3 + p04*x(2)^4;
end

function f =evaluate_localbest(x1,x2,x3)%求解粒子环形邻域中的局部最优个体
K0=[x1;x2;x3];
K1=[func(x1),func(x2),func(x3)];
[maxvalue index]=max(K1);
plocalbest=K0(index,:);
f=plocalbest;
end
```

附录 3 问题四 NSGA-3 优化算法（python）

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D  # 空间三维画图
from utils import uniformpoint,funfun,cal,GO,envselect,IGD
import copy
import random

#参数设置
N_GENERATIONS = 500                                    # 迭代次数
POP_SIZE = 100                                         # 种群大小
name = 'DTLZ2'                                         # 测试函数选择，目前可供选择
DTLZ1,DTLZ2,DTLZ3
M = 3                                                 # 目标个数
t1 = 20                                                # 交叉参数 t1
t2 = 20                                                # 变异参数 t2
pc = 1                                                 # 交叉概率
pm = 1                                                 # 变异概率

#画图部分
if(M<=3):
    fig = plt.figure()
    ax = Axes3D(fig)

##############################################################################
##############################################################################
#######
```

```
#产生一致性的参考点和随机初始化种群
Z,N = uniformpoint(POP_SIZE,M)#生成一致性的参考解
pop,popfun,PF,D = funfun(M,N,name)#生成初始种群及其适应度值，真实的 PF,自变量个数
popfun = cal(pop,name,M,D)#计算适应度函数值
Zmin = np.array(np.min(popfun,0)).reshape(1,M)#求理想点
#ax.scatter(Z[:,0],Z[:,1],Z[:,2],c='r')
#ax.scatter(PF[:,0],PF[:,1],PF[:,2],c='b')

#迭代过程
for i in range(N_GENERATIONS):
    print("第{name}次迭代".format(name=i))
    matingpool=random.sample(range(N),N)
    off = GO(pop[matingpool,:],t1,t2,pc,pm)#遗传算子,模拟二进制交叉和多项式变异
    offfun = cal(off,name,M,D)#计算适应度函数
    mixpop = copy.deepcopy(np.vstack((pop, off)))
    Zmin = np.array(np.min(np.vstack((Zmin,offfun)),0)).reshape(1,M)#更新理想点
    pop = envselect(mixpop,N,Z,Zmin,name,M,D)
    popfun = cal(pop,name,M,D)
    if(M<=3):
        ax.cla()
        type1 = ax.scatter(popfun[:,0],popfun[:,1],popfun[:,2],c='g')
        plt.pause(0.00001)

# 绘制 PF
if(M<=3):
    type2 = ax.scatter(PF[:,0],PF[:,1],PF[:,2],c='r',marker = 'x',s=200)
    plt.legend((type1, type2), (u'Non-dominated solution', u'PF'))
else:
    fig1 = plt.figure()
    plt.xlim([0,M])
    for i in range(pop.shape[0]):
        plt.plot(np.array(pop[i,:]))
plt.show()

#IGD
score = IGD(popfun,PF)
```

```
'''import numpy as np

ary=np.array([1,2,3,4,5])
print(ary,type(ary))
```

```
for i in ary:
 print(i)

print(ary[0],ary[0:5])'''

'''i=1
a=0
while i<100:
    i=i+1
    a=a+i
print(a)
a=0
for i in  range(1,100):
    a+=(i+1)
print(a)

print(sum(range(1,101)))
from functools import reduce
def add(x,y):
    return x+y
print(reduce(add,range(1,101)))

def sumFun(max):
    if max<=100 and max>=0:
        return max+sumFun(int(max)-1)
    else:
        return 0
print(sumFun(100))


def my_abs(x):
    if x>0:
        return x
    else:
        return -x


def my_ppp(x,y):
    bn=y(x)
    print(bn)

my_ppp(-5,my_abs)
```

```
def main(n):
    sum=0
    for x in range(n):
        sum+=(x+1)
    print(sum)

if __name__=='__main__':
    main(55)'''
'''递归函数'''
def fact(n):
    if n==1:
        return 1
    return n*fact(n-1)

if __name__=='__main__' :
    print(fact(1))
    print(fact(2))
```