

队伍编号	902774
题号	D

基于改进粒子群优化算法的“脱氧合金化”配料优化

摘要

我国积极推动钢铁工业转型升级，对脱氧合金化工艺环节的配料方案进行优化是技术升级的重要部分。本文结合灰色关联分析模型（GRA）和 SPSS 相关系数分析结果得到了影响 C，Mn 收得率的主要因素。利用支持向量回归模型（SVR）与贝叶斯岭回归模型对 C，Mn 元素收得率进行预测，最后利用改进粒子群模型优化了合金配料方案。

针对问题一，本文先对附件 1 的数据进行了预处理，得到了较完整的实验数据，从而计算了 C、Mn 元素的历史平均收得率，分别为 91.09%、88.39%。其次，通过运用灰色关联模型和 SPSS 相关性系数分析得到两种主要因素的分析结果。综合考虑，分析出对 C 收得率的主要影响因素为转炉终点温度、钢水净重、锰硅合金 FeMn68Si18、石油焦增碳剂、碳化硅 (55%)、硅钙碳脱氧剂；对 Mn 收得率的主要影响因素为终点温度、钢水净重、硅锰面、锰硅合金 FeMn68Si18, 锰硅合金 FeMn64Si27。

针对问题二，本文选取 80%的数据作为训练集，20%作为测试集。先利用支持向量回归（SVR）模型进行收得率预测，结果得到 SVR 模型的 C、Mn 预测准确率分别为：54%、53%。提出了两种优化方案：从数据集层面，我们调整输入数据集，使预测准确率分别提高至 55%、57%；从模型层面，采用贝叶斯岭回归模型进行预测。对比发现贝叶斯岭回归预测模型效果优于 SVR 预测模型，贝叶斯岭回归模型的预测成功率为：57%、56%。

针对问题三，本文将问题二中预测合金收得率的结果作为已知数据。为了计算最低成本的合金料配比，先假设了目标函数并添加约束条件，利用改进粒子群优化算法(PSO)进行求解。在合金收得率最优条件下，此时得到碳，锰元素最优化合金配料总成本分别为 11186.83，9764.95 元。

针对问题四，我们通过建立的模型计算出的结果与实际情况，向炼钢厂领导提出了具有一定参考价值的建议，希望能对实际生产有一些积极影响。

关键词：支持向量回归；优化粒子群；贝叶斯岭回归；相关性系数分析；灰色关联分析

目录

一、问题重述.....	1
二、模型假设.....	1
三、符号说明.....	2
四、问题分析.....	2
五、数据预处理.....	3
5.1 处理方法.....	4
5.2 处理结果.....	7
六、问题一的模型与求解.....	8
6.1 碳、锰元素历史收得率.....	8
6.2 影响收得率的主要因素.....	8
6.2.1 灰色关联预测模型.....	8
6.2.2 双变量相关性系数分析.....	10
6.3 小结.....	11
七、问题二的模型与求解.....	12
7.1 收得率预测模型—SVR.....	12
7.1.1 SVR 模型原理.....	12
7.1.2 SVR 模型求解.....	12
7.2 模型的优化—数据改进.....	14
7.3 模型的优化—模型改进.....	15
7.3.1 贝叶斯岭回归模型原理.....	15
7.3.2 贝叶斯岭回归模型结果.....	16
7.3.3 SVR 模型与贝叶斯岭回归模型预测情况的对比.....	18
7.4 小结.....	20
八、问题三的模型与求解.....	20
8.1 成本优化模型—PSO.....	20
8.1.1 PSO 模型原理.....	20
8.1.2 合金配料线性变换模型原理.....	21
8.1.3 联合模型建立.....	22
8.1.4 联合模型求解.....	23
8.2 小结.....	25
九、总结.....	25
十、模型优缺点.....	25
10.1 主要因素分析模型——灰色关联模型.....	25
10.2 收得率预测模型一——SVR.....	25
10.3 收得率预测模型二——贝叶斯岭回归模型.....	26
10.4 成本优化模型——PSO.....	26
十一、建议信（问题四）.....	27
十二、参考文献.....	28
十三、附录.....	29

一、问题重述

在钢铁冶炼中，脱氧合金化是一种重要工艺。根据需求成品钢的种类加入不同量、不同种类的合金进行转炉冶炼和脱氧合金化。通过数据对该环节建立数学模型，预测并优化投入合金的种类及数量，在保证钢水质量的同时有效地降低合金钢的生产成本，这是钢铁企业急需解决的问题。基于此种背景，建立合适的数学模型，分析并解答以下问题：

- （1）根据附件 1，计算碳、锰两种元素的历史收得率，分析影响收得率的主要因素。
- （2）在问题 1 的基础上，构建数学模型，对碳、锰两种元素的收得率进行预测，并优化模型及算法，最大限度提高这两种元素收得率的预测准确率。
- （3）根据以上预测结果及附件 2，建立数学模型，实现钢水脱氧合金化成本优化计算，给出合金配料方案。
- （4）根据以上研究结果，写一封建议信给炼钢厂领导。

二、模型假设

- 假设钢水质量在脱氧合金化前后不会发生变化
- 假设脱氧合金化过程不受工人的技术水平影响
- 假设所使用的转炉规格与质量一致
- 暂不研究时间因素对结果的影响
- 暂不研究脱氧方法的不同影响
- 暂不研究加入合金顺序的影响

三、符号说明

符号	说明
σ	标准误差项
$\xi_i(x)$	关联系数
ρ	分辨系数
r_i	关联度
$\rho(X,Y)$	Pearson 相关系数
ρ_s	Spearman 相关系数
τ	Kendall 相关系数
ξ_i	松弛因子
ε	阈值
$\phi(x)$	特征向量
$p(\omega)$	先验概率
β	数据噪声精度
ε_i	误差值

注：剩余部分未标注的符号将在文中进行进一步说明。

四、问题分析

问题一分析：解决问题一最先要做的是处理附件一数据的残缺问题，我们选择通过回归方法补齐大量空白数据。具体来说主要通过多元线性回归，以已知的连铸正样碳当量（Ceq）为因变量，通过查询文献，找出其相关的自变量，使用 SPSS 对已知数据集进行多元线性回归，得到相关的数学模型后计算未知数据；其他部分的较多空白数据，我们以简单线性回归即可拟合，剩余个别缺失我们采用均值补齐；同时，对于少量的不符合生产要求、数值波动过大的异常数据，直接剔除。经过数据的预处理后，我们得到一个较为完整的数据表，通过题目中已经给出的合金收得率的公式，就能够直接套用计算得出历史收得率。第二小问，我们选择灰色关联分析和相关性系数两种方法分别提取对收得率贡献比较大的元素：使用 Python 对数据选取合适的关联系数进行灰色关联的计

算，同时使用 SPSS 对数据进行 Pearson 和 Kendall 相关系数的检验，通过对比两个方案的结果，选择综合结果贡献最大的数据集作为影响收得率的主要因素。

问题二分析：根据问题一的结果，通过查阅转炉炼钢与脱氧合金化的相关文献，从回归预测模型的角度，自主选择 80% 的数据为训练集，选择构建支持向量回归（SVR）模型预测所求元素收得率，并以剩余的 20% 数据作为测试集，检验模型的可行性。在优化模型方面，从数据和模型两个方面进行改进：在数据上，增加影响因素、对在数据预处理中补齐的数据进行重新排列组合，再次运行得到新的收得率，并通过与先前结果的对比，比较优化情况；在模型上，选择贝叶斯岭回归模型进行重新描述，比较测试集的准确度、误差等参数，讨论模型优化的方向和成功率。

问题三分析：将问题二预测出的收得率作为已知数据，结合附录二中的价格，建立粒子群（PSO）模型来进行优化计算，其中，将总成本作为目标函数，以元素质量、元素总量、钢号的内控区间作为约束条件，结合模型求解构成目标函数最小值合金配比结果，并给出最优合金配料方案。在运行中通过计算机自主优化参数，控制迭代次数，得到全局最优解。

问题四分析：根据前文研究的结果，结合炼钢厂实际情况，写一封具有实际指导价值的建议信给炼钢厂领导。

五、数据预处理

合金收得率预测模型受 C、Mn 两种元素的转炉终点、连铸正样、加入合金配料数值的影响，由于仪表精度、现场测量环境、可靠性及人为因素的等可能出现的情况会造成误差，使得采集的数据会存在异常情况，同时，表格还特别出现了数据的大量缺失。在计算时，不仅需要剔除不完整和异常数据，还需将缺失值补齐。如果仅用原始数据来预测，会导致预测的成功率下降，输入变量众多且相互影响，则须对样本数据进行预处理，具体处理方式如下表 1：

统计情况	数据名称	判断原因	处理方法
数据异常	转炉终点温度	高于 1800°	删除
	连铸正样 N_val	整列为 NULL	删除
	连铸正样 O_val	整列为 NULL	删除
	连铸正样 B_val	绝大部分为 NULL	删除
	转炉终点 Si	少数数值为负	取绝对值
	钒铁 (FeV50-B)	整列为 NULL	删除
少量数据缺失	转炉终点温度	空值	平均值填充
	转炉终点 C, Mn, S, P, Si	空值	平均值填充
大量数据缺失	转炉终点 Mn	空值	多元线性回归
	转炉终点 P	空值	多元线性回归
	连铸正样 C, Mn	空值	多元线性回归

表 1 样本预处理

5.1 处理方法

1、缺失值处理：

常用的缺失值处理方式如下：(1)忽略元组 (2)人工填写缺失值 (3)使用一个全局常量填充缺失值 (4)使用属性的均值填充缺失值 (5)使用与给定元组属同一类的所有样本的属性均值 (6)使用最可能的值填充缺失值:可以用回归、使用贝叶斯形式化的基于推理的工具或决策树归纳确定。

本题中的处理方案:我们先对缺失最多的连铸正样 C 和连铸正样 Mn 数据进行了分析后,决定采用多元线性回归模型进行数据补全。为了确定不同因素对连铸正样的影响,我们分别对可能影响连铸正样 C 和连铸正样 Mn 的因素进行相关性分析。选取了多组与连铸正样 C 相关性较高的数据集建立回归模型。函数模型描述如下:

$$y = a + bx_{i1} + cx_{i2} + dx_{i3} + \cdots + gx_{in} + \varepsilon_i$$

其中 $x_{i1}, x_{i2}, \cdots, x_{in}$ 为自变量; a, b, \dots, n 回归系数; ε_i 为随机变量。同理可以建立连铸正样 Mn 的回归模型,模型结果如下:

$$\begin{cases} y_{SeqC} = 2.163 \times 10^{-3} + 1.269 \times 10^{-6} \times a_1 + 7.151 \times 10^{-8} \times a_2 + 1.03 \times 10^{-6} \times a_3 + 4.525 \times 10^{-10} \times a_4 \\ y_{SeqMn} = 1.004 \times 10^{-2} + 8.66 \times 10^{-6} \times b_1 + 8.022 \times 10^{-1} \times b_2 + 8.72 \times 10^{-2} \times b_3 + 3.054 \times 10^{-6} \times b_4 - 4.002 \times 10^6 \times b_5 \end{cases}$$

其中 y_{SeqC}, y_{SeqMn} 分别为连铸正样 C 和连铸正样 Mn; a_1, a_2, a_3, a_4 分别代表钒铁 (FeV50-B),

硅铝钙, 碳化硅 (55%) 和硅锰面 (硅锰渣); b_1, b_2, b_3, b_4, b_5 分别代表钒氮合金 (进口), 连铸正样 Si, 连铸正样 Cr, 硅铝合金 FeAl30Si25, 硅铁 FeSi75-B; 以上两个回归模型的确定系数分别为 0.26, 0.6。将数据上述模型即可补全数据。

函数模型描述如下: 同样有较大数据缺失的是转炉终点的 P 和 Mn 元素的含量, 通过多次尝试我们发现, 转炉终点的值与上部分计算出的与其对应的连铸正样量有线性关系, 通过简单线性回归则可以轻松得到转炉终点的 P、Mn 含量。

$$\begin{cases} y_P = 0.0001 - 0.1023 \times c_1 + 0.2012 \times c_2 + 0.1435 \times c_3 - 0.0063 \times c_4 \\ y_{Mn} = 0.0005 + 0.3639 \times d_1 - 0.8357 \times d_2 + 1.7855 \times d_3 + 0.0556 \times d_4 \end{cases}$$

其中 y_P, y_{Mn} 分别为转炉终点 P 和转炉终点 Mn; c_1, c_2, c_3, c_4 分别代表转炉终点 C, 转炉终点 Mn, 转炉终点 S, 转炉终点 Si; d_1, d_2, d_3, d_4 分别代表转炉终点 C, 转炉终点 S, 转炉终点 P, 转炉终点 Si; 以上两个回归模型的确定系数分别为 0.37, 0.47。将数据上述两个回归模型联立即可补全数据

相对于缺失量极大的数据, 只有个别数据缺失的一些元素, 我们则选择直接采用同钢号的其他炉号的该列取算术平均数代替。

值得注意的是, 在进行回归计算上, 由于不同钢号的钢对元素的需求量不同, 我们将不同钢号的数据分开进行回归, 以期达到更大的精度。具体描述如下:

不同的钢号意味着其本身蕴含有不同的元素。这些元素将在脱氧合金化过程中起到不同的作用, 也会对脱氧合金化之前的钢水产生影响。

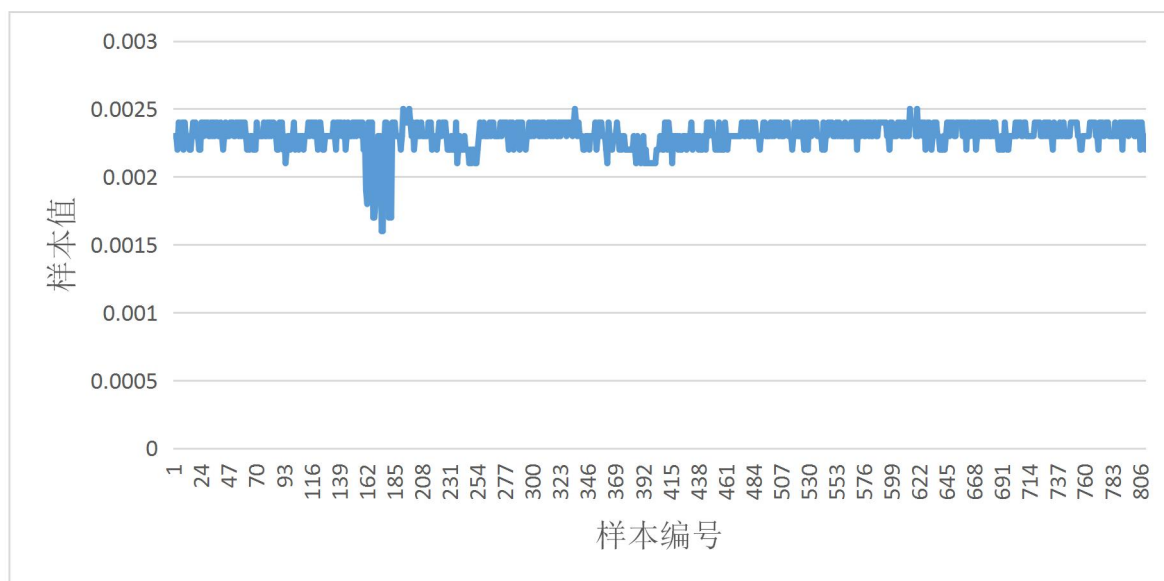


图 1 样本连铸正样 C 含量折线图

如图所示在附录一已给出连铸正样 C 数值趋于平稳，但在数值区间[166, 199]明显低于均值，对这部分数据进行分析，发现由于钢号为 HRB400B 与钢号为 Q345B 碳含量的有着明显的区别。绘制出两种钢号对比的折线图，图例中系列 1 代表 HRB400B，系列 2 为 Q345B。

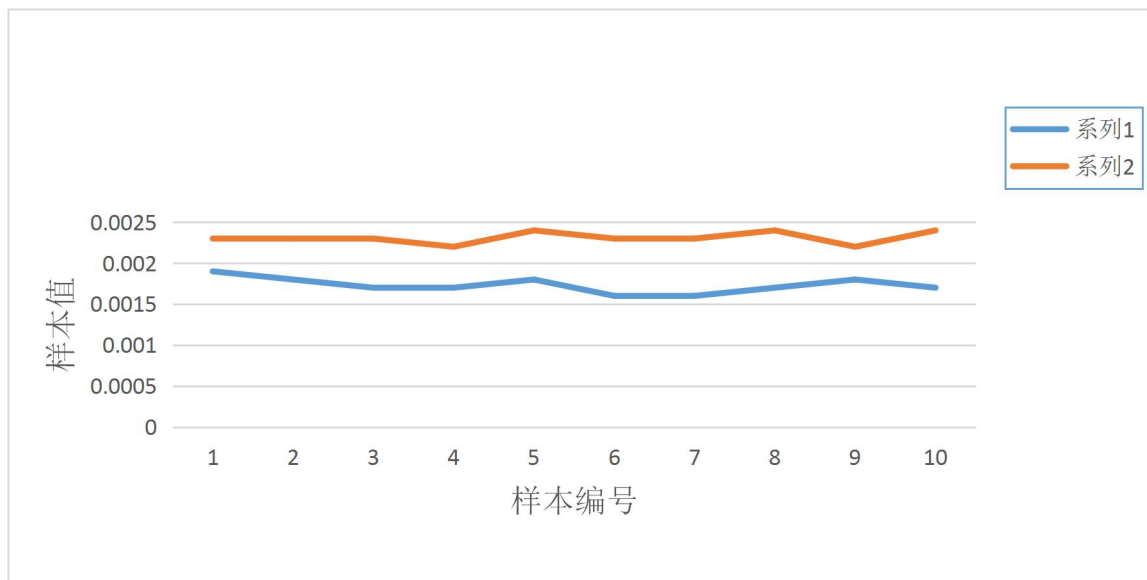


图 2 异种钢号连铸正样 C 含量对比图

2、数据异常：

异常值剔除的方法采用的是拉依达准则，即 3σ 准则：对于部分特殊数据，当偏差大于 3σ 的时候，该数据为异常值，需要被及时剔除， σ 的计算公式为：

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

x_i 指检测值的原始数据， \bar{x} 指检测值的平均值，样本容量 $n = 1677$ 。

当偏差大于 3σ 的时候，该数据为异常值，异常值的判断准则如下：

$$|x_i - \bar{x}| > 3\sigma$$

处理本文数据时，我们对整列为 NULL，或者几乎整列为 NULL 的连铸正样 N、B、O 元素，进行整列删除，经查询发现，转炉炼钢时会进行脱氧脱氮反应，最后成品钢水几乎不含这些元素，因此可以直接整列去掉。对于有部分负值的转炉终点 Si，显然这些负值不符合原则，我们对偏差过大的数据直接去除，对于偏差较小的数据采用了绝对值计算。本数据同时出现了操作导致温度过高造成的错误数据，在验证时，这些过高温的炉号中的各元素含量明显出现了不同程度的误差，但由于数据量较少，本文就直接做去除处理。

3、标准化处理：

将数据按照一定规则缩放，使之落入一个小的特定区间。这样去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。本文将数据标准化处理以方便后续计算。主要方法如下：

(1) Min-Max 标准化即离差标准化，是对原始数据的线性变换，使结果落到 $[0, 1]$ 区间，转换函数如下：

$$y_i = \frac{x_i - \min\{x_j\}}{\max\{x_j\} - \min\{x_j\}} \quad (1 \leq i, j \leq n)$$

其中 $\max\{x_j\}$ 为样本数据的最大值， $\min\{x_j\}$ 为样本数据的最小值。

(2) Z-score 标准化即标准差标准化，经过处理的数据符合标准正态分布，即均值为 0，标准差为 1，其转化函数为：

$$y_i = \frac{x_i - \bar{x}}{s} \quad (1 \leq i \leq n)$$

其中 \bar{x} 为所有样本数据的均值， s 为所有样本数据的标准差。

(3) 归一标准化：

$$y_i = \frac{x_i}{\sum_i^n x_i^2} \quad (1 \leq i \leq n)$$

若新序列 $y_1, y_2, \dots, y_n \in [0, 1]$ 无量纲，则显然有结果 $\sum_1^n y_i = 1$ 。

由于原始数据存在不同的量纲单位，因此数据的大小差别非常大，数据的范围也不相同。差异的较大会增大某些变量对预测结果的影响，同时会减弱某些变量对预报模型的影响，所以需要对剔除异常值后的数据进行归一化。将所有数据都转化为 $[0, 1]$ 之间的数，可以消除数据因大小不一而造成的偏差，最后对预测结果进行反归一化还原。

5.2 处理结果

在对样本数据进行预处理之后，我们提取了处理前的已知数据以及预处理后获得的新数据。根据下表 2 可知，主要有 7 类元素的数据在数据集中发生重大变化。我们将分别对每类元素处理前后数据的进行了比较。我们选取了基本统计量的均值、标准偏差、标准平均值误差以及范围作为比较量。

且由统计结果可知，每类元素的统计量数值相近，表明了数据预处理之后的数据集具有一定的可靠性，可以为后续问题的解决提供支持。处理这些元素在同钢号的情况下的数据，运用 SPSS 分析如下：

报告							
	转炉终点Mn	转炉终点P	连铸正样C	连铸正样Mn	连铸正样S	连铸正样P	连铸正样Si
平均值	.001017	.000220	.002309	.013355	.000300	.000252	.003763
数字	250	250	810	810	810	810	810
标准偏差	.0002670	.0000759	.0001013	.0004526	.0000385	.0000539	.0003794
标准平均值误差	.0000169	.0000048	.0000036	.0000159	.0000014	.0000019	.0000133
范围	.0016	.0005	.0009	.0027	.0002	.0003	.0030

表 2—1 样本原始数据描述性统计表

报告							
	转炉终点Mn	转炉终点P	连铸正样C	连铸正样Mn	连铸正样S	连铸正样P	连铸正样Si
平均值	.001051	.000230	.002306	.013313	.000296	.000251	.003875
数字	1676	1676	1676	1676	1676	1676	1676
标准偏差	.0002973	.0000720	.0000863	.0003963	.0000476	.0000505	.0005131
标准平均值误差	.0000073	.0000018	.0000021	.0000097	.0000012	.0000012	.0000125
范围	.0017	.0004	.0009	.0027	.0004	.0003	.0030

表 2—2 样本预处理数据描述性统计表

六、问题一的模型与求解

6.1 碳、锰元素历史收得率

合金收得率是评判炼钢工艺的重要指标，其在脱氧合金化工艺中的定义为吸收的合金元素质量与加入该元素总量之比。依此定义并结合附录 1，得到如下计算元素 X 的收得率公式：

$$X\text{金属收得率} = \frac{(\text{转炉终点}X - \text{连铸正样}X) \times \text{钢水净重}}{\text{加入物质中所含}X\text{元素的质量}}$$

将预处理好的数据集导出为 CSV 格式，运用 Python 语言进行程序设计，代入计算。最后分别计算出数据集中碳、锰元素的历史平均收得率约为 84.239% 与 86.033%，详见附录 1。

收得率分布特征\元素	C	Mn
平均值	0.910947862	0.883967187
标准偏差	0.074655425	0.051662935
范围	0.94092915	0.289577565
标准平均值误差	0.002114113	0.001463005
峰度	5.182	0.242
偏度	-1.044	-0.542

表 3 历史收得率

6.2 影响收得率的主要因素

6.2.1 灰色关联预测模型

(1) 灰色关联分析原理

灰色关联是一种基于灰色系统的寻求各因素之间关系的模型。在使用灰色关联之前，需要在数据集中确定参考序列。其余数据将组成比较数列。由于，不同的数据具有量纲单位，所以需要先进性数据的标准化处理。假设序列 $y_i(x) = (y_i(1), y_i(2), \dots, y_i(n))$ 为比较数列。采用均值化法处理比较数列公式表示如下：

$$y_i(x) = \frac{y_i(x)}{y_i}, i = 1, 2 \dots n; x = 1, 2 \dots m$$

在解决了量纲问题之后，将进行最为核心的一步，即计算关联系数 $\xi_i(x)$ ，假设 y^* 为参考数列可以得到

$$\xi_i(x) = \frac{\min_i \min_x |y^*(x) - y_i(x)| + \rho \max_i \max_x |y^*(x) - y_i(x)|}{|y^*(x) - y_i(x)| + \rho \max_i \max_x |y^*(x) - y_i(x)|}$$

其中 $\rho \in (0, \infty)$ 是分辨系数，本次取值为 0.5。关联系数是反应数列之间在某个点的关联程度。因为想要进行整体性比较需要通过关联度 r_i 进行解决。公式如下：

$$r_i = \frac{1}{n} \sum_{x=1}^n \xi_i(x), x = 1, 2 \dots n$$

最后对关联度 r_i 进行排序即可以得到比较序列与参考数列的关联程度。

(2) 灰色关联分析结果

我们将数据中连铸正样去除，以剩余数据作为该模型的输入参数，使用 Python 进

行编程实现（见附录），最后分析影响结果如下：

★C 收得率降序排列：

钢水净重、转炉终点温度、碳化硅、锰硅合金、转炉终点 S、石油焦增碳剂、转炉终点 Mn、转炉终点 C。

★Mn 收得率降序排列：

转炉终点温度、锰硅合金、钢水净重、转炉终点 S、石油焦增碳剂、转炉终点 Mn、转炉终点 C、碳化硅、转炉终点 P、转炉终点 Si。

（3）影响因素详解

①转炉终点温度：脱氧过程中，在加入合金元素之后，会立刻发生吸热反应，温度的高低将直接影响分子之间的运动。实际上，脱氧与合金化可以近似理解为同时进行，温度对最终的收得率有着很大的影响。

②合金料：有些合金的加入是作为脱氧剂，如硅铁、锰铁等等。不同的脱氧剂对于脱氧合金化的过程有着不同的氧化效果。还有部分合金是作为增碳剂、催化剂，推动化合反应的正向进行，加快反应速率。并且合金对生产中的器具也有会不同程度的影响。通过对比发现，在同一条件下，加入的合金料中含有合金的质量越多，则该元素的收得率也会更高。

③钢号：不同的钢号意味着成钢中蕴含有不同种类的元素、不同比例的元素。在制成钢中对不同元素的吸收量和需求量也不同，影响收得率的炉温、喷溅情况等因素也不同，那么最终得到的收得率也有很大差别，但同种钢号的合金收得率值则较为稳定。

④C、Mn、S、P、Si 的含量：磷 P，磷对钢有害的元素，它极易使钢变得冷脆；硫 S，硫也是钢中的有害元素，使钢的强度降低；碳 C：能增加钢的冷脆性和时效敏感性；锰 Mn，在炼钢过程中，锰是良好的脱氧剂和脱硫剂；硅 Si，在炼钢过程中加硅作为还原剂和脱氧剂，所以镇静钢含有 0.15—0.30%

6.2.2 双变量相关性系数分析

（1）相关性系数模型原理

三个相关性系数（Pearson, Spearman, Kendall）反应的都是两个变量之间变化趋势的方向以及程度，其值范围为 $[-1, 1]$ ，如果为 0 则表示两个变量不相关，正值表示正相关，负值表示负相关，值越大相关性越强。

①Pearson 矩相关系数：正态分布数据

$$\rho(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_x \sigma_y}$$

分母是变量的标准差，不能为 0，则两个变量中任何一个的值不能都是相同的。如果相同，用 Pearson 相关系数没办法算出这个变量与另一个变量之间的相关性。

②Spearman 秩相关系数：离散有序变量

$$\rho_s = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

对两个变量(X,Y)的数据进行排序，然后记下排序以后的位置为秩次，构造有序的对数据后计算其矩相关系数。

③Kendall 秩相关系数：连续性有序分类变量

$$\tau = \frac{2}{n(n-1)} \cdot \sum_{1 \leq i < j \leq n} \text{sgn}(x_j - x_i)(y_j - y_i)$$

(2) 相关性系数分析结果

★C 收得率：选取 Pearson 系数计算。

相关性						
		C	转炉终点C	连铸正样C	转炉终点温度	钢水净重
C	Pearson 相关性	1	.105**	-.135**	-.043	-.146**
	显著性（双尾）		.000	.000	.081	.000
	N	1676	1676	1676	1676	1675

表 4—1 C 元素收得率相关性

相关性								
		C	Zscore(石油 焦增碳剂)	Zscore: 碳化 硅(55%)	Zscore(硅钙 碳脱氧剂)	Zscore: 锰硅 合金 FeMn68Si18 (合格块)	Zscore: 锰硅 合金 FeMn64Si27 (合格块)	Zscore: 硅铁 FeSi75-B
C	Pearson 相关性	1	.136**	.216**	.009	.101**	-.006	.023
	显著性（双尾）		.000	.000	.705	.000	.794	.349
	N	1676	1676	1676	1676	1676	1676	1676

表 4—2 C 元素收得率相关性

由上表 4 可知，C 收得率影响因素降序排列有：碳化硅、钢水净重、石油增碳剂、转炉终点 C、锰硅合金（FeMn68Si18）、转炉终点温度、硅铁、硅钙碳脱氧剂、锰硅合金（FeMn64Si27），前五项的累积贡献率达到 84%。

★Mn 收得率：选取 Kendall 系数计算。

相关性							
			Mn	转炉终点Mn	连铸正样Mn	转炉终点温度	钢水净重
肯德尔 tau_b	Mn	相关系数	1.000	-.064**	-.012	.012	.016
		显著性（双尾）	.	.000	.480	.462	.328
		N	1676	1676	1676	1676	1675

表 5—1 Mn 收得率相关性

相关性								
			Mn	Zscore: 锰硅合金 FeMn68Si18 (合格块)	Zscore: 锰硅合金 FeMn64Si27 (合格块)	Zscore: 硅铁 (合格块)	Zscore(硅锰面 (硅锰渣))	Zscore(硅铝合金 FeAl30Si25)
肯德尔 tau_b	Mn	相关系数	1.000	-.187**	.005	-.068**	.314**	.018
		显著性（双尾）	.	.000	.799	.001	.000	.357
		N	1676	1676	1676	1676	1676	1676

表 5—2 Mn 收得率相关性

由上表 5 可知，Mn 收得率影响因素降序排列有：硅锰面、锰硅合金（FeMn68Si18）、硅铝合金、硅铁、转炉终点 Mn、硅铝锰合金球、钢水净重、转炉终点温度、锰硅合金（FeMn64Si27），其中，前五项累积贡献率有 71.4%。

6.3 小结

在对问题一的求解上，我们首先通过数据的预理解决了所求元素 C、Mn 的历史收得率，后通过灰色关联预测和相关性系数分析分别计算了不同元素收得率的主要影响因素，综合得出如下结果：C 收得率：转炉终点温度, 钢水净重, 转炉终点 C, 锰硅合金 FeMn68Si18(合格块), 石油焦增碳剂, 碳化硅(55%), 硅钙碳脱氧剂; Mn 收得率：硅锰面(硅锰渣)，锰硅合金 FeMn68Si18(合格块), 锰硅合金 FeMn64Si27(合格块)。

七、问题二的模型与求解

7.1 收得率预测模型—SVR

7.1.1 SVR 模型原理

支持向量机是常用的机器学习算法。应用在回归分析上，被称为支持向量回归，回归问题要求解的结果是连续的。在回归中，我们通过建立一个数学模型，使得线性函数 $f(x)$ 尽可能逼近与结果 y ，对于模型输出的结果与真实结果 y 的误差就会被当作损失，有且只有两者相同时，损失为零。但这在实际运用中是基本不肯能实现的。而在 SVR 中

是可以容忍 $f(x)$ 与 y 之间的偏差的，由此我们设置一个阈值 ε ，当损失的绝对值大于该阈值时，我们才认为有损失。这也可以认为增大了预测结果被判断是正确的可能。我们引入松弛因子 ξ_i, ξ_i' 将该问题表示如下约束：

$$\begin{cases} f(x_i) - y_i \leq \varepsilon + \xi_i \\ y_i - f(x_i) \leq \varepsilon + \xi_i' \\ \xi_i, \xi_i' \geq 0 \end{cases}$$

同时我们可以设 $f(x)=w^T+b$ 并引入拉格朗日乘子法获得一个拉格朗日函数 $L(w, b, \partial, \partial', \xi, \xi', \mu, \mu')$ 并对 ξ_i, ξ_i', w, b 偏导为零最后加上 KKT 条件，可求得 SVR 的解形如下：

$$f(x) = \sum_{i=1}^m (\partial_i - \partial_i') x_i^T x + b$$

此时可以推知 $\partial_i - \partial_i' \neq 0$ 的数据必在阈值之外。使用 $\phi(x)$ 表示特征向量则可表示为

$$f(x) = w^T \phi(x) + b$$

使用 $\kappa(x, x_i)$ 代表核函数，最终可以推得 SVR

$$f(x) = \sum_{i=1}^m (\partial_i - \partial_i') \kappa(x, x_i) + b$$

7.1.2 SVR 模型结果

★C 收得率：如下图 3 是 SVR 模型下 C 收得率的预测结果，由图可见预测值的趋势基本能够拟合真实值曲线，但经计算发现，模型的精确值只有 50-54% 左右，截取的部分结果图中我们也能发现，这种模型拟合还是与真实值有一定出入。

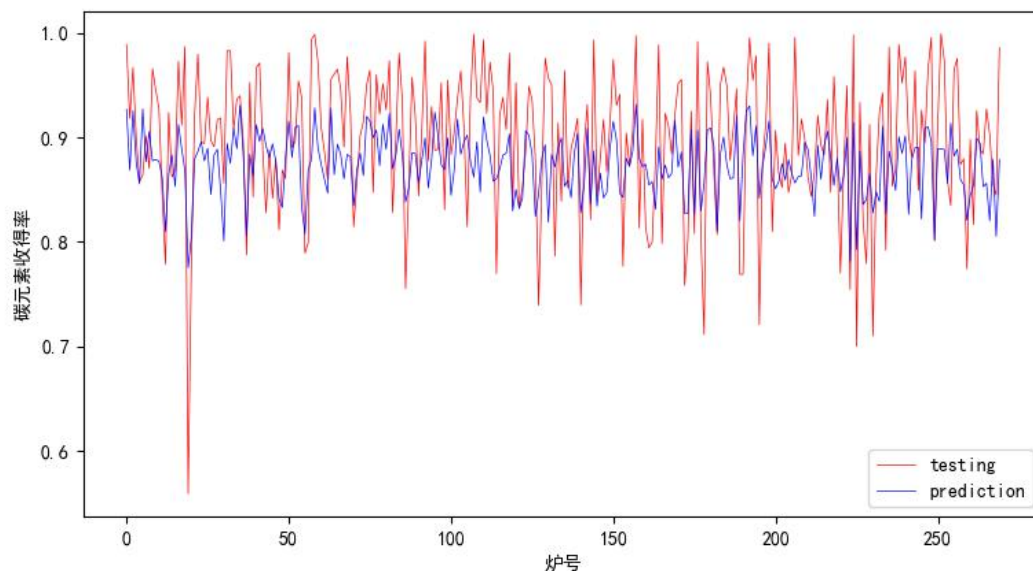


图 3—1 SVR 模型 C 元素收得率测试集预测结果

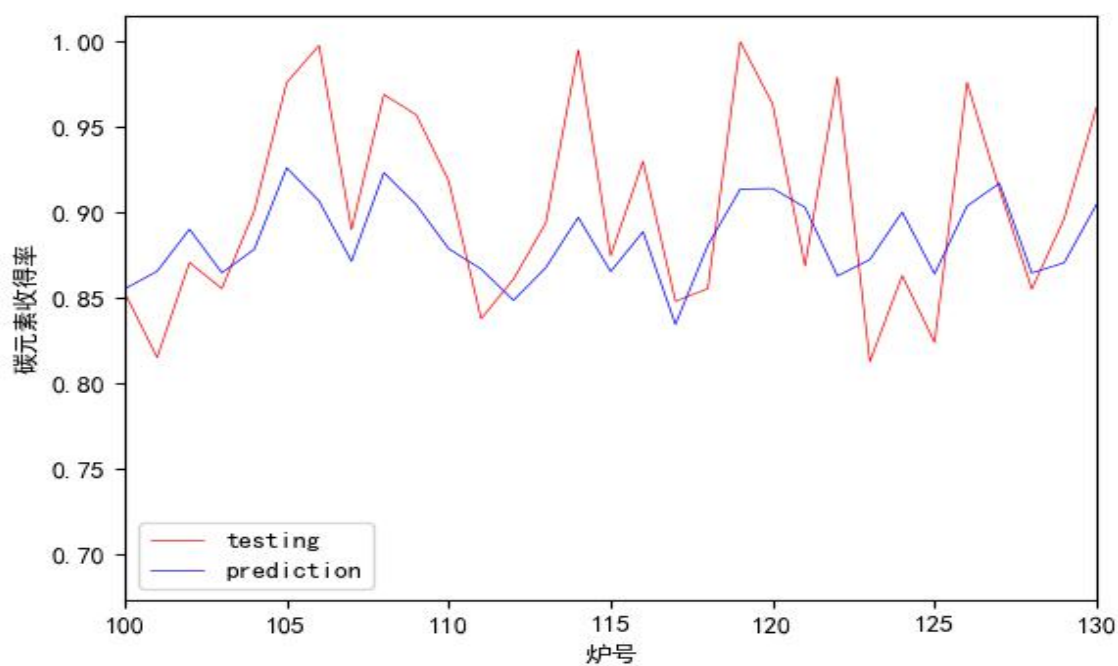


图 3—2 SVR 模型 C 元素收得率测试集预测部分结果

★Mn 收得率：如下图 4 是 SVR 模型下 Mn 收得率的预测结果，下峰值拟合情况较好，上峰值拟合情况一般，准确率大约在 53-54%之间，拟合结果要优于 C 收得率，但总体拟合程度一般。

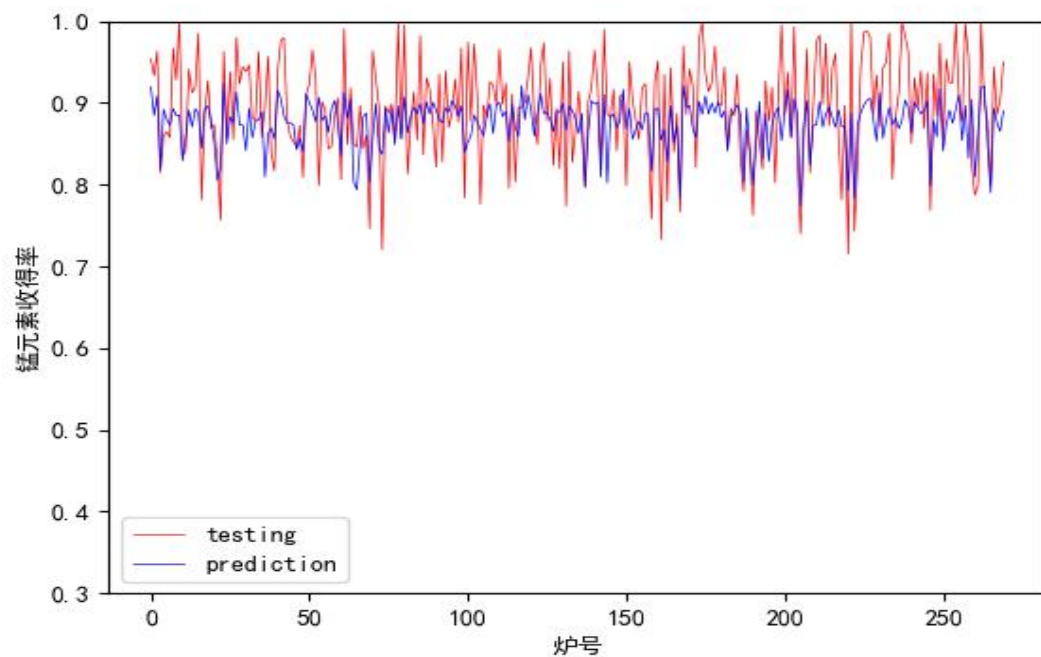


图 4—1 SVR 模型 Mn 元素收得率测试集预测结果

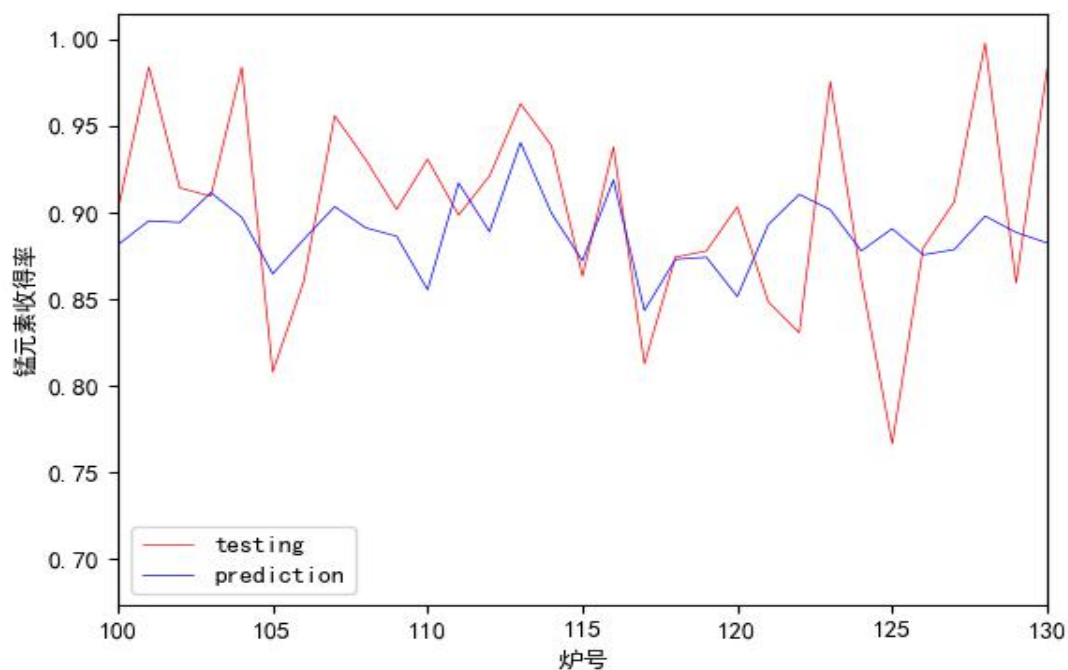


图 4—2 SVR 模型 Mn 元素收得率测试集预测部分结果

7.2 模型的优化——数据改进

SVR 模型进行计算时，我们选取了第一问中计算出的合金收得率影响因素的前五个主要影响因素，由第一问我们可以知道，C、Mn 收得率前五个主要影响因素的累积贡献率分别达到了 80%和 70%，那么，在进行数据优化时，我们适当增加了三个自变量，使

得影响因子的累计贡献率达到 86%和 78%，此时再次进行运算后可以发现准确率有明显提升，说明对数据进行适当改进是可行的。结果如下表 6：

但同时可以发现，这种数据改进方法十分有局限性，自变量的个数是有限的。并且如果采用其他数据改进方案，如对数据预处理中的数据进行重新排列组合，则很大可能会出现数据与数据不匹配，计算得到负值的情况，反而准确率有所下降甚至无法计算结果。

碳元素收得率	SVR 预测准确率(%)	增加自变量的 SVR 预测准确率(%)
score	54.306	54.588
锰元素收得率	SVR 预测准确率(%)	增加自变量的 SVR 预测准确率(%)
score	53.733	56.956

表 6 优化数据集准确率对比表

7.3 模型的优化——模型改进

7.3.1 贝叶斯岭回归模型原理

贝叶斯线性回归的引入主要是由于最大似然估计中很难决定模型的复杂程度，岭回归加入惩罚参数目的也是解决这个问题的，同时还可以采用数据的标准化处理，因此后续计算上将使用标准化数据进行分析。

设参数为 ω ，则参数的先验概率就是 $p(\omega)$ ，设数据噪声的精度是 β ，也就说误差值

$\varepsilon_i = (y_i - \omega^T x_i)$ 服从 $N\left(0, \frac{1}{\beta}\right)$ ，且已知。那么似然函数就是 $p = (t|\omega)$ ， t 生成的目标 y 值。

当然条件中隐含了参数 X ，那么作如下设定：

$$p(\omega) = N(\omega|m_0, S_0)$$

(m_0, S_0 分别是正态分布的均值和协方差)

那么根据贝叶斯公式 $P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$ 有

$$p(\omega|t) = N(\omega|m_N, S_N)$$

其中参数为 m_N ， S_N^{-1} 计算公式如下：

$$\begin{cases} m_N = S_N (S_0^{-1} m_0 + \beta \phi^T t) \\ S_N^{-1} = S_0^{-1} + \beta \phi^T \phi \end{cases}$$

由于先验分布是高斯，似然估计是高斯分布，所以后验概率同样也是高斯分布

假设 ω 的分布有一个精度参数 ∂ 来确定，也就是 $p(\omega|\partial) = N(\omega|0, \partial^{-1}I)$ ，那么对应的后验分布的参数为 $m_N = S_N \beta \phi^T t$ ， $S_N^{-1} = \partial I + \beta \phi^T \phi$

那么，将这个后验分布写成具体的概率密度函数，然后对 ω 求对数，就得到了如下公式：

$$\ln p(\omega|t) = -\frac{\beta}{2} \sum \{t_n - \omega^T \phi(x_n)\}^2 - \frac{\partial}{2} \omega^T \omega + \text{const}$$

预测分布 (predictive distribution)：归纳所有的 ω ，则

$$p(t|t, \alpha, \beta) = \int p(t|\omega, \beta) p(\omega|t, \alpha, \beta) d\omega$$

$$p(t|t, \alpha, \beta) = N(t|m_N^T \phi(x), \sigma_N^2(x))$$

其中有， $\sigma_N^2(x) = \frac{1}{\beta} + \phi(x)^T S_N \phi(x)$

7.3.2 贝叶斯岭回归模型结果

★C 收得率：如图 5 为使用贝叶斯岭回归求得的 C 收得率预测结果，相对于 SVR，明显岭回归的图像更加贴近原曲线波动情况，在部分值中有部分重叠的情况，明显准确率更高，大约在 57-63%之间。

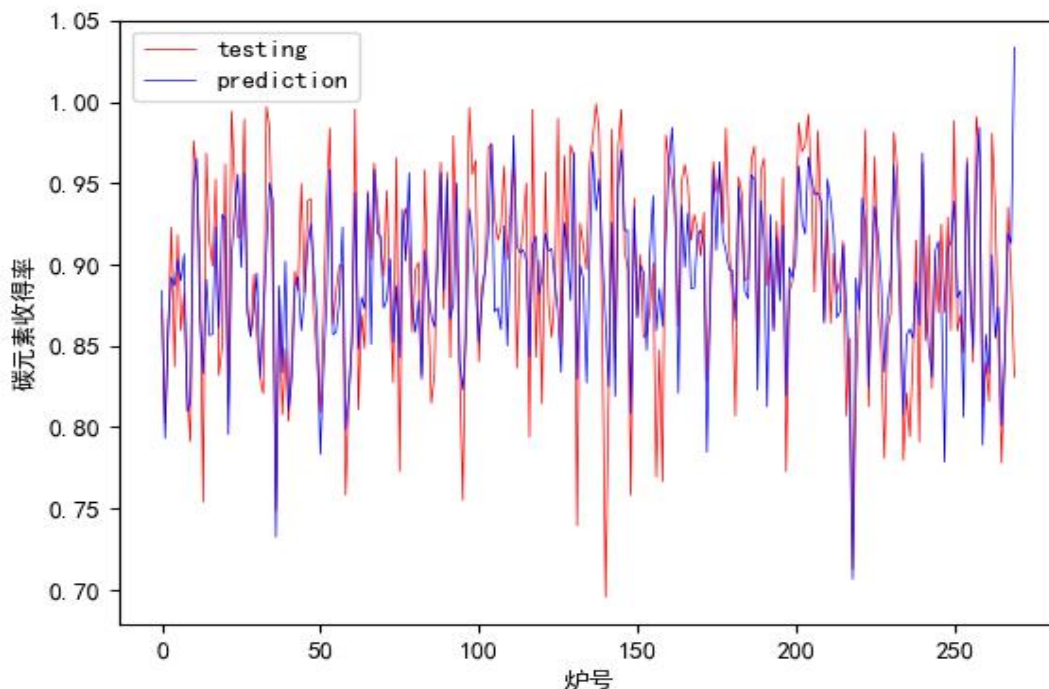


图 5—1 贝叶斯岭回归 C 元素收得率测试集预测结果

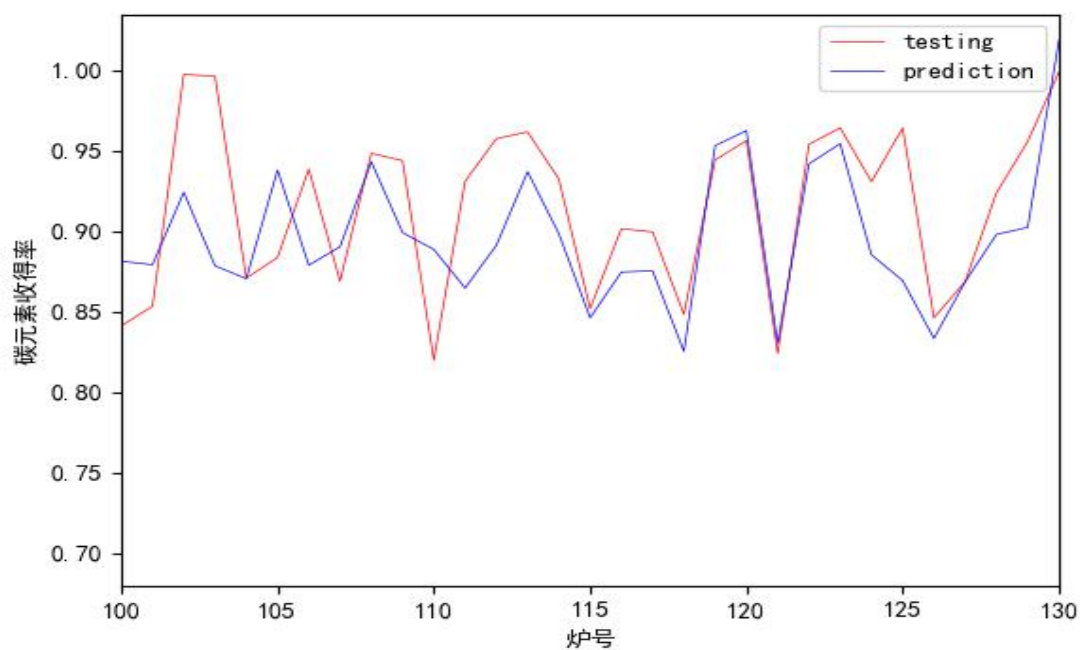


图 5—2 贝叶斯岭回归 C 收得率测试集部分预测结果

★Mn 收得率：如图 6 为贝叶斯岭回归模型下的 Mn 收得率的预测结果，拟合程度也明显优于 SVR 的图像，准确率大约在 56-60%之间，去除了 SVR 值时上峰值大部分不能拟合的情况。

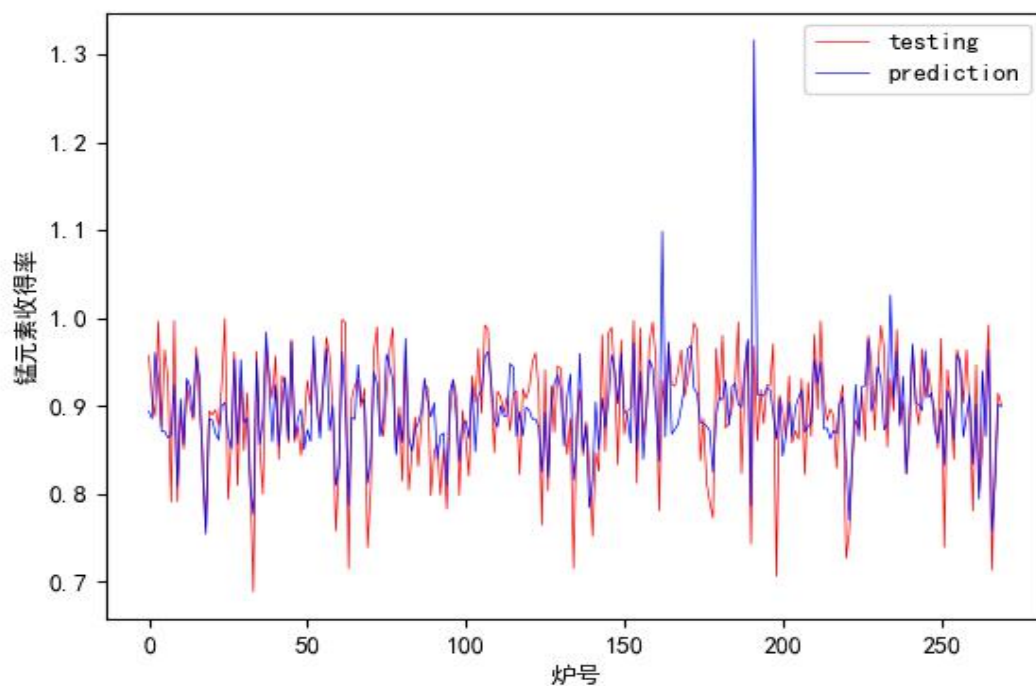


图 6—1 贝叶斯岭回归 Mn 收得率测试集预测结果

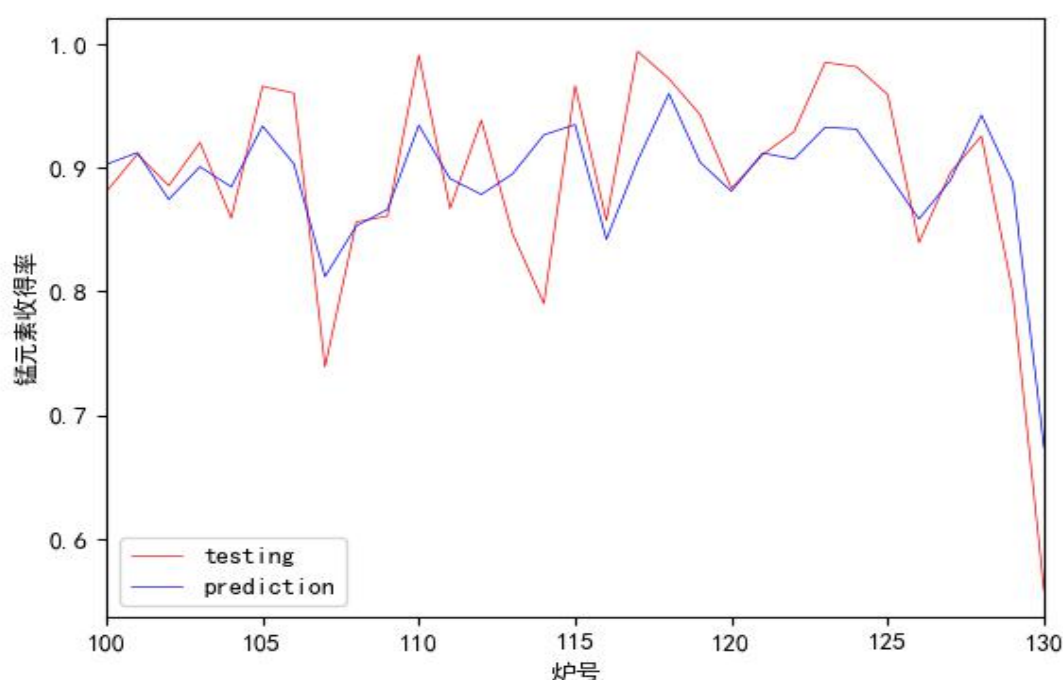


图 6—2 贝叶斯岭回归 Mn 收得率测试集部分预测结果

7.3.3 SVR 模型与贝叶斯岭回归模型预测情况的对比

(1) 交叉验证法

在进行比较前，我们引入交叉验证法：它的基本思想是将原始数据进行分组，一部分做为训练集来训练模型，另一部分做为测试集来评价模型。它用于评估模型的预测性能，尤其是训练好的模型在新数据上的表现，可以在一定程度上减小过拟合。主要方法如下：

①留出法 (holdout cross validation)

在机器学习任务中，拿到数据后，我们首先会将原始数据集分为三部分：训练集、验证集和测试集。训练集用于训练模型，验证集用于模型的参数选择配置，测试集对于模型来说是未知数据，用于评估模型的泛化能力。这个方法操作简单，只需随机把原始数据分为三组即可。不过如果只做一次分割，它对训练集、验证集和测试集的样本数比例，还有分割后数据的分布是否和原始数据集的分布相同等因素比较敏感，不同的划分会得到不同的最优模型，而且分成三个集合后，用于训练的数据比较少。

②k 折交叉验证 (k-fold cross validation)

k 折交叉验证通过对 k 个不同分组训练的结果进行平均来减少方差，因此模型的性能对数据的划分就不那么敏感。步骤如下：

第一步，不重复抽样将原始数据随机分为 k 份。

第二步，每一次挑选其中 1 份作为测试集，剩余 k-1 份作为训练集用于模型训练。

第三步，重复第二步 k 次，这样每个子集都有一次机会作为测试集，其余机会作为训练集。在每个训练集上训练后得到一个模型，用这个模型在相应的测试集上测试，计算并保存模型的评估指标。

第四步，计算 k 组测试结果的平均值作为模型精度的估计，并作为当前 k 折交叉验证下模型的性能指标。

（2）对比结果说明

★C 收得率：如下表 7，对于 C 元素，贝叶斯回归预测值明显更贴近真实值结果，并且准确率也明显优于 SVR。

序号	炉号	碳元素实际收得率	SVR 预测值	贝叶斯岭回归预测值
1	7A05515	0.863793655	0.900419335	0.858597585
2	7A05514	0.955156951	0.87744526	0.901042758
3	7A05513	0.943843604	0.845731826	0.921779863
4	7A05512	0.915990686	0.895030335	0.878320524
5	7A05511	0.860704169	0.899598718	0.876647765
6	7A05510	0.889723895	0.866535357	0.880410602
7	7A05509	0.964182143	0.897242496	0.958103655
8	7A05508	0.898296013	0.880180671	0.858377923
9	7A05507	0.851903695	0.857649517	0.851329717
10	7A05506	0.830565945	0.821465617	0.864324983

表 7-1 C 元素收得率预测值表

碳元素收得率	SVR 预测准确率 (%)	贝叶斯岭回归预测准确率 (%)
交叉验证法 score	48.025	56.093
score	54.306	57.981

表 7-2 C 收得率预测值准确率表

★Mn 收得率：如下表 8，由交叉验证法可以得到，Mn 的收得率准确率增量，说明使用 SVR 时，合金收得率预测结果准确率不够稳定，说明使用贝叶斯岭回归的方案更优。

序号	炉号	锰元素实际收得率	SVR 预测值	贝叶斯岭回归预测值
1	7A05515	0.890713219	0.860733069	0.88243508
2	7A05514	0.869901962	0.834191901	0.881421727

3	7A05513	0.929454661	0.880301227	0.958524316
4	7A05512	0.885287303	0.906421303	0.85365916
5	7A05511	0.848090723	0.862558378	0.856932785
6	7A05510	0.862019272	0.81755323	0.83523949
7	7A05509	0.930713219	0.870571057	0.949849475
8	7A05508	0.946448403	0.90927115	0.917657806
9	7A05507	0.925352363	0.814939408	0.866391137
10	7A05506	0.930241312	0.821359725	0.909243262

表 8-1 Mn 元素收得率预测值表

锰元素收得率	SVR 预测准确率(%)	贝叶斯岭回归预测准确率(%)
交叉验证法 score	33.111	58.283
score	53.733	55.66

表 8-2 Mn 元素收得率预测值准确率表

7.4 小结

在解决问题二时，我们首先提出了 SVR，即支持向量回归，来作为拟合预测的模型，并选取了 80%数据作为训练集，20%为测试集，得出了合金元素收得率的预测图像。同时我们发现，这种方法预测结果准确率较低，我们采取数据优化和模型优化两种方法对原模型的改进，进行了自变量数据增加和贝叶斯岭回归模型分析的优化操作，使得预测值的稳定性和准确性有了显著增加。

八、问题三的模型与求解

8.1 基于 PSO 的成本优化模型

8.1.1 PSO 模型原理

粒子群优化算法(PSO)是模拟鸟群行为的模型，主要目的是使粒子能够找到向解空间，并能找到最优解。同遗传算法类似，PSO 也是一种基于群体迭代的，但并没有遗传算法用的交叉以及变异，而是粒子在解空间追随最优的粒子进行搜索。PSO 的优势在于简单，容易实现，无需梯度信息，参数少，适合于处理实优化问题。具体 PSO 算法流程如下：

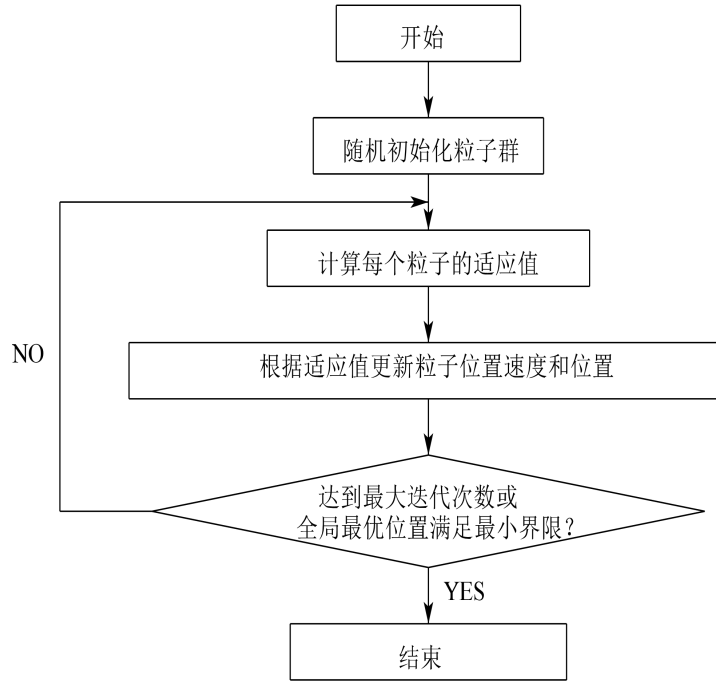


图 7 粒子群优化算法图解

在粒子群算法中，每次迭代粒子的速度 v_i 与位置 x_i 更新公式如下^[3]：

$$\begin{cases} v_i^{t+1} = \mu v_i^t + c_1 r_1 (pbest_i - x_i^t) + c_2 r_2 (gbest_i - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{cases}$$

其中， μ 为惯性系数，取值在 $[0, 8-1.2]$ ； t 表示迭代次数，一般取 $0.9 \sim 0.4$ 线性递减值； c_1 ， c_2 是加速因子，通常取 $c_1 = c_2 = 2$ ； r_1 ， r_2 在 $(0, 1)$ 之间随机取值； $pbest$ 表示个体极值，记录单个粒子发现的最好的位置； $gbest$ 表示全体极值，记录整个种群经历过的最好的位置。

8.1.2 合金配料线性变换模型原理

假设合金成本为 Z 。在约束条件下，寻求合金配料的最低成本就是最终目标。目标函数如下：

$$\min Z = \sum_{i=1}^n c_i x_i$$

其中 c_i 表示第 i 种合金的成本， x_i 代表第 i 种合金的加入量。

为了求解这个目标函数，我们对合金配料与收得率之间的关系进行了研究。参考文献，具体约束条件如下：

$$\begin{cases} \sum_{i=1}^n (x_i \times r_j \times m_{ij}) + G \times E_j \geq \left[G + \sum_{i=1}^n x_i \right] \times E_{\min j} \\ \sum_{i=1}^n (x_i \times r_j \times m_{ij}) + G \times E_j \leq \left[G + \sum_{i=1}^n x_i \right] \times E_{\max j} \end{cases}$$

G 对应附录一的钢水净重; E_j 为附录一中 j 类元素在添加物料前的含量 (可通过转炉终点计算); $E_{\min j}$ 与 $E_{\max j}$ 分别代表在某类钢产品中 j 类元素含量的内控区间的最小值与最大值; r_j 代表 j 类元素的收得率由第二问的模型预测得出的结果; m_{ij} 为第 i 种配料中 j 类元素的含量。

8.1.3 联合模型的建立

通过 PSO 模型前期运行结果发现, 当惯性系数 μ 较大时, 全局收敛能力强, 当惯性系数 μ 较小时, 则局部收敛能力较强。因此, 在迭代过程中, 如果能让惯性系数 μ 随着迭代的次数线性改变, 使得初惯性系数 μ 取值相对较大, 全局搜索能力较强。并且随着迭代次数的增加, 惯性系数 μ 慢慢的减小, 这样就能够保证在进入最优的区域时, 粒子群算法有着相对比较强的局部搜索能力。惯性系数 μ 线性变换的计算公式如下:

$$\mu = (\mu_1 - \mu_2) \frac{MaxDT - DT}{MaxDT} + \mu_2$$

其中, μ_1 / μ_2 为惯性系数的起始值和最终值, $MaxDT / DT$ 为迭代次数的最大值和当前迭代次数。

根据 PSO 模型和线性变换模型, 我们进行组合联立, 建立了基于线性变换的粒子群变化。参数更新有:

c1	c2	w	r1	r2	pN	dim	max_iter
2	2	0.9(线性递减)	random(0, 1)	random(0, 1)	20	5	30

表 9 优化粒子群模型参数

由上述惯性系数的新计算公式与 PSO 模型, 我们得到的算法流程更新如下:

步骤一: 在约束条件内, 初始化粒子群并随机产生粒子后计算 $pbest$ 、 $gbest$ 。

步骤二: 根据公式计算每次迭代后的粒子位置、速度和惯性系数。

步骤三: 由适应函数计算适应度并与 $pbest$ 比较, 选取最优结果。

步骤四: 判断 $pbest$ 与 $gbest$ 的结果, 选取最优。

步骤五：判断是否满足终止条件，若满足，则输出结果；不满足则回到步骤二。

如下图 8 为算法变化前后的一个适应函数适应度曲线变化：明显变化后适应曲线更优。

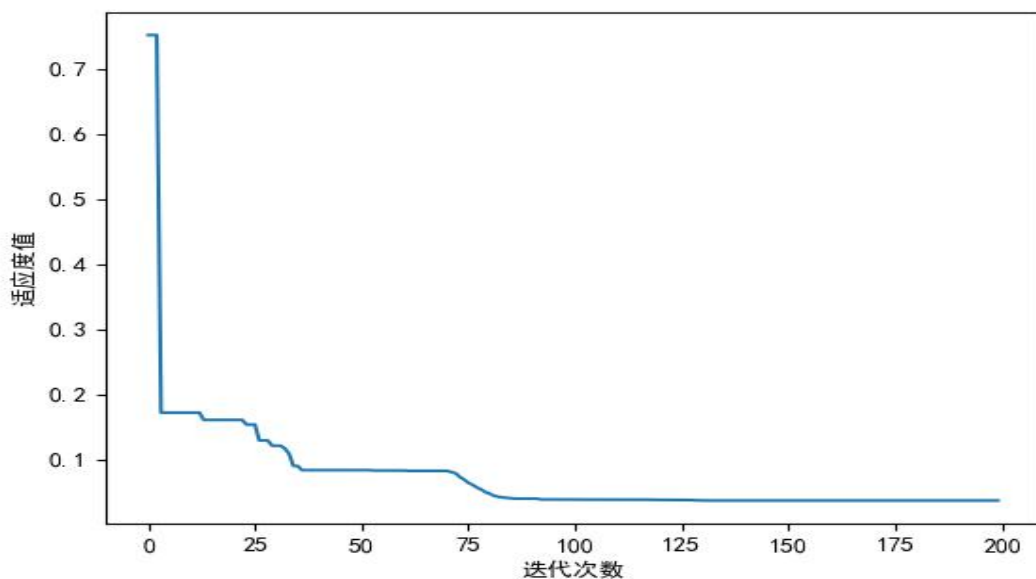


图 8-1 标准粒子群算法适应度曲线

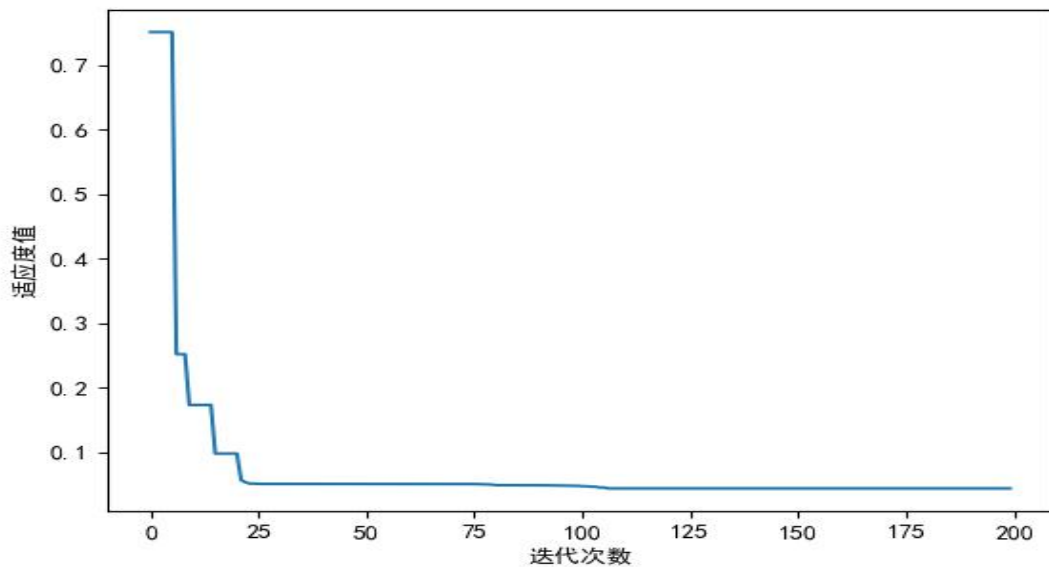


图 8-2 线性变换粒子群算法适应度曲线

8.1.4 联合模型的求解

根据更新后的算法，我们求解得到目标函数与输入变量的关系，即在归一化后合金配料成本单价与收得率分别在 $[0, 1]$ 区间范围内变化，并以此为输入变量，预测目标函数值（总成本）的变化情况，如图 9。选取三维图最低点，则为该适应函数下最低成本。最后由此点即可确定合金配料的配比情况。

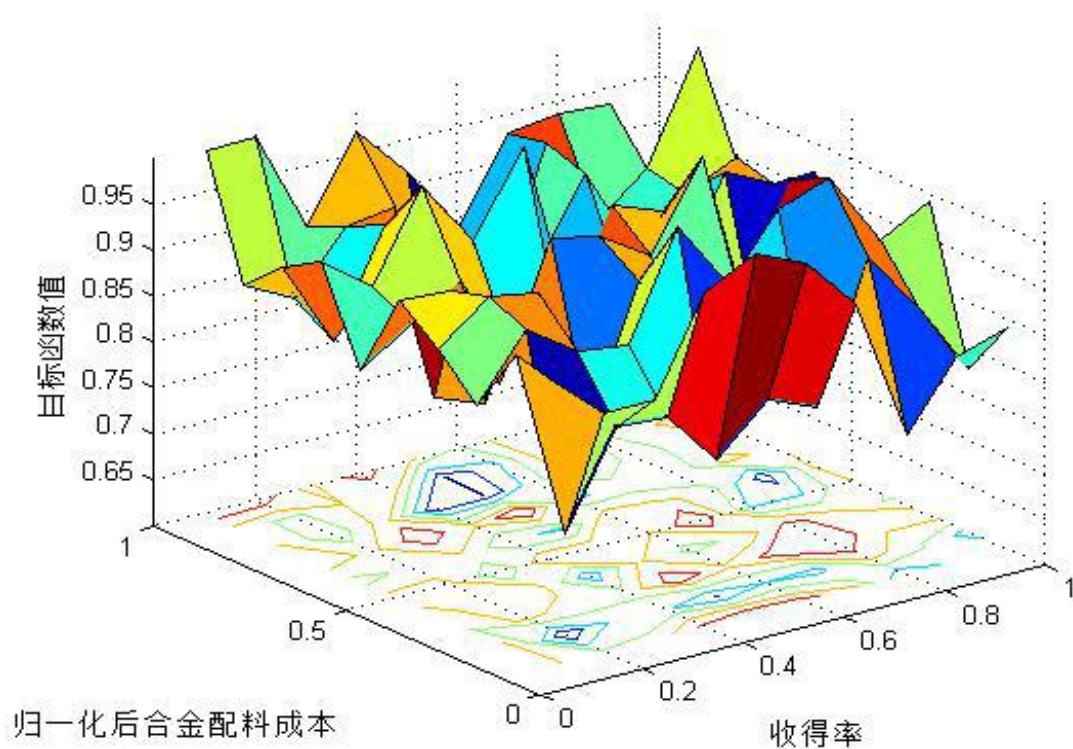


图 9 收得率、合金配料成本与目标函数值的关系图

由上图 9 中最低点，我们能够对应得到该适应函数下的最佳配料比，如下表 10，我们通过对比发现在元素收得率范围一定的情况下，测试集中经验配方的成本要明显高于优化配方，且收得率较低，说明改进后粒子群优化算法明显减小了合金剂的成本，有效的实现了高收得率低成本的实验要求。

合金配料	锰硅合金 FeMn68Si18(合格块)	石油焦增碳剂	碳化硅 (55%)	硅钙碳脱氧剂	成本	C 元素收得率
经验配方	1437	76	104	2.7	12706.35	88%
优化配方	1248	90	96	4	11186.83	89%

表 10-1 C 元素收得率的合金配料方案

合金配料	硅锰面（硅锰渣）	锰硅合金 FeMn68Si18(合格块)	锰硅合金 FeMn64Si27(合格块)	成本	Mn 收得率
经验配方	200	1062	16	10305.7	91%
优化配方	220	973	20	9764.95	91.60%

表 10-2 Mn 元素收得率的合金配料方案

8.2 小结

解决问题三的过程中，我们选用了多目标优化的 PSO 模型，即粒子群优化算法，我们不仅在迭代粒子的速度上改变了惯性系数的范围，避免收敛速度过快。而且同时将线性模型与 PSO 模型有效结合，提出了基于线性变换的粒子群优化算法，使得惯性系数随着迭代次数的增加而不断减小，以此来尽量求取全局最优的情况。最后由得到符合约束条件的目标函数（成本最小），得到收得率一定时的最低成本及该情况下的合金配料比。

九、总结

首先，我们发现附录一的数据有大量缺失现象，其中还有异常值。缺失数据如果直接选择删除的话，那么数据分析的效果偏差将是巨大的。因此我们利用多种方式进行补齐。当有了较为完整的数据集之后，我们根据题目所给的金属收得率公式进行了计算。附录一中所给的元素很多，只采用一种关联分析方法，难以确保分析结果的正确性。因此我们采用了灰色关联模型和计算相关性系数两种方式分析出主要影响因素。

其次，我们建立了 SVR 预测模型和贝叶斯岭回归模型开始对 C、Mn 的收得率进行预测。通过调整数据集，得到更加准确的收得率预测值。通过对比发现贝叶斯岭回归模型效果最好。因此将这个算法预测出的收得率作为后续合金配料模型的输入数据。

最后，我们设计目标函数与约束条件，用优化粒子群算法来求解出合金配料方案。本次研究内容耦合度高，如果有任何一部分分析效果比较差，将影响后续的所有结果。限于时间紧张和水平有限，研究结果还存在不足。如果有足够的时间和精力，我们将预测其他元素的收得率并综合考虑元素之间的相互影响，进一步提高预测模型与优化模型的准确性。

十、模型优缺点

10.1 主要因素分析模型——灰色关联模型

优点：灰色关联模型容错性小，精度较高，简单实用且易操作。

缺点：对历史数据有很强的依赖性，做大量数据预测时，误差较大。

10.2 合金收得率预测模型——SVR

优点：SVR 模型简单，参数较少，它能够较好地处理非线性和高维问题，并且它的核函

数具有多样性，可以应对不同种的数据。从理论上来说，SVR 模型算法得到的是全局最优解，可以有效解决神经网络的局部最优问题，并且具有较好的推广性。

缺点：SVR 模型以支持向量机为基础，不可避免的会出现在解决大规模样本时计算困难，并且针对多分类的样本难以实施，只能选择进行多个二分类组合，较为耗时。

10.3 合金收得率预测模型二——贝叶斯岭回归模型

优点：贝叶斯岭回归模型在训练数据时更加简易快速，岭回归模型的惩罚系数会减小数据过拟合的情况，并且模型灵活，容易调整。

缺点：惩罚系数的设置极易造成过度拟合的情况，使预测结果准确率下降，并且对数据的相关性要求苛刻，很难进行校准。

10.4 成本最优模型——PSO

优点：PSO 模型易于求解连续性问题，可以自主有效的优化系统参数，以便更快地逼近最优解。

缺点：局部寻优能力较差，在处理多峰问题时，容易产生早熟收敛的现象。

十一、建议信（问题四）

尊敬的各位领导：

我们非常荣幸有机会能与各位领导探讨脱氧合金化这项工艺的配料问题。炼钢市场的竞争是激烈的，如何在保证钢铁质量的同时又降低成本一直是钢铁市场热议的话题。基于前文研究得到的结果，我们冒昧地提出了一些对炼钢中脱氧合金化这一步骤的建议。不足之处，还请领导批评指正。

一、规范脱氧合金化工艺流程

脱氧合金化流程中的每步操作对最终的炼钢结果都会有不同的影响。我们采用了灰色关联分析、计算了 Pearson 和 Kendall 的相关系数来分析影响收得率主要因素。综合分析结果，我们发现一些相关因素，例如，转炉终点温度, 钢水净重, 转炉终点 C 等对炼钢结果有重大影响。因此，规范化工工艺流程，尽可能减少各项指标的人工和计算误差是非常重要的。

贵厂可以制定标准化的生产标准，强化员工技术水平，合理更新工厂设备，以减少脱氧合金化工艺流程中成品钢水受到的影响。

二、应用合金自动化配料优化模型

自动化配料模型一直被广泛应用于生产之中。该类模型能科学地解决配料问题，精细配料的准度，减少生产成本。合金在炼钢成本中占有率高达 15%。如果企业能对合金配料进行合理优化，在不影响成品质量的情况下降低材料成本，那么企业在市场的竞争力将极大提高，企业的整体经济利益也会受到积极的影响。因此应用合金自动化配料模型对于现代化炼钢企业是非常有必要的。经过详细的研究和试验，我们建立了一种钢水脱氧合金化的优化成本控制优化模型。该模型基于已知的历史数据，通过粒子群优化成本，得到最优的合金配比。我们希望该模型能够满足炼钢厂的实际需求，降低炼钢的成本。我们希望贵厂能应用该模型在实际生产中，并对企业的生产起到积极的作用。

以上就是我们的提出的建议。诚然，我们的模型是还存在局限性，离商业化应用还有很大距离。但仍然希望我们的建议能得到您们的认可。感谢您们在百忙之中阅读我们的建议。祝您身体健康，工作顺利，企业蓬勃发展，日胜一日！

此致

敬谢

Mathorcup 团队成员

十二、参考文献

- [1]韩敏, 徐俏等, 基于收得率预测模型的转炉炼钢合金加入量计算, 炼钢, 第 26 卷第 1 期: 44-47, 2010。
- [2]徐喆, 毛志忠, 钢包精炼炉元素收得率的影响因素分析及预报, 钢铁, 第 47 卷第 3 期: 34-37, 2012。
- [3]马志远, 合金元素收得率多重回归分析的建模, 钢铁技术, 第 4 期: 13-16, 2000。
- [4]韩敏, 徐俏等, 基于微粒群算法的转炉合金加入量多目标优化模型, 控制与决策, 第 25 卷第 12 期: 1901-1904, 2010。
- [5]韩敏, 赵耀等, 基于鲁棒相关向量机的转炉炼钢终点预报模型, 控制理论与应用, 第 28 卷第 3 期: 343-350, 2011。
- [6]袁平, 王福利等, 电弧炉炼钢合金成分控制模型, 热能工程, 第 35 卷第 2 期: 1-4, 2006。
- [7]陈晓东, 鲍可进等, 模糊线性规划在转炉合金模型中的应用, 江苏大学学报(自然科学版), 第 23 卷第 1 期: 66-70, 2002。
- [8]李强, 基于改进粒子群算法的艾萨炉配料优化, 昆明理工大学, 2015。
- [9]司守奎, 李玺, 数学建模算法与应用, 北京: 国防工业出版社, 2011。
- [10]陈绍新, 多目标优化的粒子群算法及其应用研究, 大连: 大连理工大学, 2007。

十三、附录

C 与 Mn 收得率计算程序(在软件 pycharm 运行)

```
1. import pandas as pd
2. import csv
3. data = pd.read_csv('dataOne.csv', encoding='GBK')
4. Data = data.values
5. Cm = []
6. Cz = []
7. Mnm = []
8. Mnz = []
9. #计算钢水中合金元素质量
10. for i in range(len(Data)):
11.     a = float(Data[i][7])*(Data[i][8]-float(Data[i][2]))
12.     b = float(Data[i][7]) * (Data[i][9] - float(Data[i][3]))
13.     Cz.append(a)
14.     if(i<len(Data)):
15.         Mnz.append(b)
16. #计算加入的合金元素总质量
17. for i in range(len(Data)):
18.     Cm.append(Data[i][16]*0.031 + Data[i][17]*0.031 + Data[i][19]*0.00374 +
19.         Data[i][21]*0.017
20.         + Data[i][22]*0.00006 + Data[i][23]*0.00006 + Data[i][24]*0.96 +
21.         Data[i][25]*0.017
22.         + Data[i][26]*0.017 + Data[i][27]*0.3 + Data[i][28]*0.225692308)
23.     if(i<len(Data)):
24.         Mnm.append(Data[i][21]*0.3 + Data[i][22]*0.664 + Data[i][25]*0.664
25.             +Data[i][26]*0.664)
26. C_rate = []
27. Mn_rate = []
28. num = len(Data)
29. for i in range(num-9):
30.     if(Mnm[i] == 0):
31.         del Mnm[i]
32.         del Mnz[i]
```



```

30.
31. #计算 C,Mn 元素的每次的可得率
32. for i in range(len(Cz)):
33.     C_rate.append(Cz[i]/Cm[i])
34.     if (i < len(Mnz)):
35.         Mn_rate.append(Mnz[i]/Mnm[i])
36. num = len(C_rate)
37. #去除异常数据
38. for i in range(1600):
39.     if (C_rate[i] >= 1 or C_rate[i] <= 0):
40.         C_rate[i] = 0.0
41.         num -= 1
42.
43.
44. #计算 C 元素的均收得率
45. sum2 = 0.0
46. for k in range(1600):
47.     sum2 += C_rate[k]
48. C_yield = sum2/num
49. print('碳元素的收得率: ', C_yield)
50.
51. #计算 Mn 元素的平均收得率
52. num = len(Mn_rate)
53. for j in range(1600):
54.     if(Mn_rate[j] >= 1 or Mn_rate[j] <= 0):
55.         Mn_rate[j] = 0.0
56.         num -= 1
57. sum1 = 0.0
58.
59. for k in range(1600):
60.     sum1 += Mn_rate[k]
61. Mn_yield = sum1/num
62. print('锰元素的收得率: ', Mn_yield)

```

灰色关联分析程序(在软件 pycharm 运行)

```
1. import pandas as pan
2.
3. #数据预处理 初始化
4. x=pan.read_csv('mn.csv')
5. x=x.iloc[:,:].T
6. c=0.5
7.
8. # 均值化处理
9. x_mean=x.mean(axis=1)
10. for i in range(x.index.size):
11.     x.iloc[i,:]= x.iloc[i,:]/x_mean[i]
12.
13. # 提取子队列和母队列
14. pa=x.iloc[0,:]
15. ps=x.iloc[1:,:]
16.
17. # 比较队列与参考队列相减
18. t=pan.DataFrame()
19.
20. for j in range(ps.index.size):
21.     temp=pan.Series(ps.iloc[j,:]-pa)
22.     t=t.append(temp,ignore_index=True)
23.
24. #求最大差和最小差
25. maxx=t.abs().max().max()
26. minn=t.abs().min().min()
27. ksi=((minn+c*maxx)/(abs(t)+c*maxx))
28.
29. #关联度
30. r=ksi.sum(axis=1)/ksi.columns.size
31.
32. #关联度排序
33. result=r.sort_values(ascending=False)
```

34. print(result)

基于线性变化的粒子群优化算法.程序(在软件 **pycharm** 运行)

```
1. # coding: utf-8
2. import numpy as np
3. import random
4. from math import cos, sqrt
5. from operator import mul
6. from functools import reduce
7. import matplotlib
8. import matplotlib.pyplot as plt
9. # -----PSO parameter settings-----
10.
11.
12. class PSO03():
13.     def __init__(self, pN, dim, max_iter):
14.         self.w = 0.9
15.
16.         self.c1 = 2
17.         self.c2 = 2
18.         self.r1 = random.uniform(0, 1)
19.         self.r2 = random.uniform(0, 1)
20.         self.pN = pN # The number of particle
21.         self.dim = dim # Search dimension
22.         self.max_iter = max_iter # Iterations times
23.         self.X = np.array([[0.8, 0.69, 0.5], [0.6, 0.82, 0.89], [0.89, 0.82, 0.83], [0.86, 0.8,
0.84],
24.                             [0.85, 0.81, 0.84], [0.83, 0.81, 0.82], [0.84, 0.82, 0.82], [0.83, 0.92,
0.91],
25.                             [0.84, 0.82, 0.82], [0.83, 0.84, 0.82]])
26.         self.V = np.array([[0, 1, 1], [0.58, 0.84, 0.56], [0.3, 0.14, 0.3], [0.53, 0.31,
0.53],
27.                             [0.56, 0.56, 0.56], [0.5, 0.5, 0.5], [0.43, 0.43, 0.43],
[0.49, 1, 0.49],
28.                             [0.43, 0.43, 0.43], [0.5, 0.43, 0.31]])
```

```

29.         self.f = np.zeros((50, 10))
30.         self.pbest = np.zeros((self.pN, self.dim))    # Best position and global best
               position for individual experience
31.         self.gbest = np.zeros((1, self.dim))
32.         self.p_fit = np.zeros(self.pN)    # Historical best fitness value for each
               individual
33.         self.fit = 1e10    # Global optimal fitness value
34.
35.         # -----Objective function adaptive value evaluation evaluation
               function-----
36.         def evaluation(self, individual):
37.             """evaluation test objective evaluation.
38.             .. list-table::
39.                 :widths: 10 50
40.                 :stub-columns: 1
41.                 * - Type
42.                   - minimization
43.                 * - Range
44.                   - :math:`x_i \in [-600, 600]`
45.                 * - Global optima
46.                   - :math:`x_i = 0, \forall i \in \{1 \dots N\}`
                     :math:`f(\mathbf{x}) = 0`
47.                 * - evaluation
48.                   - :math:`f(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \`
49.                     \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1`
50.
51.             .. plot:: code/benchmarks/evaluation.py
52.                 :width: 67 %
53.             """
54.             return 1.0 / 4.0 * sum(x ** 2 for x in individual) - \
55.                 reduce(mul, (cos(x / sqrt(i + 1.0)) for i, x in enumerate(individual)), 1)
               + 1,
56.
57.
58.         # -----initialization-----

```

```

59.     def init_Population(self):
60.         for i in range(self.pN):
61.             self.pbest[i] = self.X[i]
62.             tmp = self.evaluation(self.X[i])[0]
63.             self.p_fit[i] = tmp
64.             if (tmp < self.fit):
65.                 self.fit = tmp
66.                 self.gbest = self.X[i]
67.             # -----Update particle position-----
68.     def iterator(self):
69.         self.w -= 0.01
70.         fitness = []
71.         g = []
72.         for t in range(self.max_iter):
73.             x = []
74.             for i in range(self.pN):
75.                 temp = self.evaluation(self.X[i])[0]
76.                 x.append(temp)
77.                 if (temp < self.p_fit[i]):
78.                     self.p_fit[i] = temp
79.                     self.pbest[i] = self.X[i]
80.                     if (self.p_fit[i] < self.fit):
81.                         self.gbest = self.X[i]
82.                         self.fit = self.p_fit[i]
83.             for i in range(self.pN):
84.                 self.V[i] = self.w * self.V[i] + self.c1 * self.r1 * (self.pbest[i] -
self.X[i]) + \
85.                     self.c2 * self.r2 * (self.gbest - self.X[i])
86.                 self.X[i] = self.X[i] + self.V[i]
87.             fitness.append(self.fit)
88.             # print(self.fit)
89.         print(self.gbest)
90.         times = []
91.         for i in range(self.max_iter):
92.             times.append(i)

```

```
93.         print(len(fitness))
94.         plt.plot(times,fitness)
95.         print(fitness)
96.         return fitness
97.
98.
99.     # -----Program execution-----
100.
101.
102.if __name__ == '__main__':
103.    my_pso = PSO03(pN=10, dim=3, max_iter=50)
104.    my_pso.init_Population()
105.    fitness = my_pso.iterator()
```