

队伍编号	904604
题号	(D)

钢水“脱氧合金化”配料方案的优化

摘 要

炼钢过程中的脱氧合金化是钢铁冶炼中的重要工艺环节，脱氧合金化中的合金收得率与炼钢成本息息相关。本文参考附件数据并查阅大量关于转炉冶炼相关资料、数据，构建了脱氧合金化模型，根据大量历史数据，求得合金元素的收得率，并对合金配料问题进行了优化。

针对问题一，首先对附件一数据进行了预处理：分别把与 C 和 Mn 合金收得率的计算有关数据中的异常数据剔除，根据钢种钢号把钢产品进行分类。根据收得率的计算思想建立了一般收得率的计算模型，对 C、Mn 的收得率进行计算。发现其中有收得率异常的数据（部分 C 收得率大于 1），这是因为在进行脱氧合金化时，会发生电极增碳和钢液回锰现象^[1]。在考虑电极增碳量和钢水中含氧量对 C 收得率的影响下，建立了基于参考炉次的自学习模型对 C 的收得率进行优化，得到 C 的平均历史收得率为 84.29%、Mn 的平均历史收得率为 91.18%。最后利用灰色关联度分析的方法对可能影响 C、Mn 收得率的因素进行分析，分别选取了几个相关性较大的因素：影响 C 收得率的因素为：转炉终点温度、转炉终点 C、转炉终点 S、转炉终点 Si、钒铁(FeV50-B)、锰硅合金、碳化硅(55%)；影响 Mn 收得率的因素为：转炉终点温度、转炉终点 C、转炉终点 S、转炉终点 Si、硅铝合金 FeAl30Si25、石油焦增碳剂、锰硅合金。

针对问题二，在问题一中我们已经算出了历史收得率和影响收得率主要因素的相关性，这里我们选取相关性较高的因素来预测其收得率。我们利用 BP 神经网络来对收得率进行预测，将影响收得率的主要因素作为输入层，层数为 7，收得率作为输出层，层数为 1。预测出 C、Mn 的收得率，并得出预测值与实际值得误差。最后我们利用粒子群算法^[3]来优化 BP 神经网络，来提高其预测准确性，并减少其拟合预测误差。

针对问题三，我们根据问题二中预测的合金收得率及附件 2，建立最小成本模型^[1]。查阅资料发现 5 种主要元素中，S、P 是有害元素，必须达到国家标准，因此在用浓度约束合金配料用量时，不予考虑，仅考虑余下三种元素。选用 6 种主要合金配料进行线性规划约束。利用 Linprog()函数解得合金配料的最佳使用量，此时成本减少 12.59%。再选取达标的炉次进行分析，算得平均减少成本 10.32%。

针对问题四，结合前面数据分析及模型计算结果，从连铸正样 Si 区间不达标和最小优化成本方面给炼钢厂领导写信发出建议，并提出加强钢铁质量把关和根据最小成本优化模型对配料进行优化的建议。

关键词：数据处理 灰色预测 BP 神经网络 最小成本模型 单纯形法

目录

一、问题重述	1
1.1 问题背景	1
1.2 问题描述	1
二、问题分析	1
2.1 问题一的分析	1
2.2 问题二的分析	2
2.3 问题三的分析	2
三、模型假设	2
四、符号说明	2
五、问题一的模型建立与求解	4
5.1 数据预处理	4
5.2 C、Mn 收得率计算模型	5
5.3 基于参考炉次法的自学习优化模型（针对 C 元素收得率）	6
5.3.1 C 收得率影响因素	6
5.3.2 模型建立	6
5.3.3 优化模型求解 C 收得率	7
5.4 灰色关联度分析	8
5.4.1 数据的选取	8
5.4.2 建立灰色关联度分析模型	9
5.4.3 模型求解	9
六、问题二的模型建立与求解	10
6.1 模型准备	10
6.2 BP 神经网络模型建立	11
6.3 模型的求解	13
6.4 模型的改进	14
七、问题三模型的建立与求解	16
7.1 模型准备	16
7.2 模型建立	16
7.3 模型求解	18
7.4 模型结果	19
八、问题四 致领导的一封信	21
九、模型优缺点	22
9.1 模型优点	22
9.2 模型缺点	22
十、参考文献	23
附录	23
附录 1 问题一 Matlab 相关代码	23
附录 2 问题二 Matlab 相关代码	26
附录 3 问题三 Matlab 相关代码	39

一、问题重述

1.1 问题背景

在国家号召减少产能过剩的大背景下，钢铁行业急需提高高附加值钢种的产量以适应当前行业环境。炼钢过程中的脱氧合金化是钢铁冶炼中的重要工艺环节。对于不同的钢种在熔炼结束时，需加入不同量、不同种类的合金，以使其所含合金元素达标，最终使得成品钢在某些物理性能上达到特定要求。如何通过历史数据对脱氧合金化环节建立数学模型，预测合金元素收得率，合理配料，并以此降低钢水脱氧合金化成本，是一个十分有价值的问题。

1.2 问题描述

建立合适的数学模型，分析并解答以下问题：

问题 1:钢水脱氧合金化主要关注 C、Mn、S、P、Si 五种元素的含量，请根据附件 1 计算 C、Mn 两种元素历史收得率，并分析影响其收得率的主要因素。

问题 2:在问题 1 的基础上，构建数学模型，对 C、Mn 两种元素收得率进行预测，并进一步改进模型及算法，尽可能提高这两种元素收得率的预测准确率。

问题 3:不同合金料的价格不同，其选择直接影响钢水脱氧合金化的成本。请根据问题 2 中合金收得率的预测结果及附件 2，建立数学模型，实现钢水脱氧合金化成本优化计算，并给出合金配料方案。

问题 4:请根据你们的研究结果，给炼钢厂领导写一封建议信（一页以内）。

二、问题分析

2.1 问题一的分析

问题一要求我们计算 C、Mn 两种元素历史收得率，并分析影响其收得率的主要因素。首先根据数据所给的信息，将钢号种类，合金加入种类等信息分类组合并进行数据处理。根据合金收得率的公式和预处理后的数据分别计算出 C、Mn 两种元素历史收得率，发现 C 的部分收得率大于 1。经查阅资料发现，其原因可能与精炼加热过程中的电极增碳导致的，另外合金收得率还与钢水和渣中的氧量密切相关，因此在考虑电极增碳和氧量的影响下，建立基于参考炉次的自学习模型对 C 的收得率进行了优化，将 C 的收得率大部分优化到 1 以下。然后采用灰色关联度分析对可能影响 C、Mn 收得率的因素进行了分析，选取了几个相关性较大的因素。

2.2 问题二的分析

问题二要求利用问题一的求解结果，即影响收得率的主要因素，构建数学模型，对 C、Mn 两种元素收得率进行预测，并进一步改进模型计算法，尽可能提高这两种元素的收得率的预测准确性。在问题一中我们已经算出了收得率与影响收得率的主要因素的相关性，这里我们选取相关性较高的因素来预测其收得率。我们利用 BP 神经网络来对收得率进行预测，将响收得率的主要因素作为输入层，收得率作为输出层，得到收得率的预测值。最后我们利用粒子群算法来优化 BP 神经网络，来提高其预测准确性，并减少其拟合预测误差。

2.3 问题三的分析

从附件 1 中可以看到，对于数据量最多的低合金 HRB400B，它需要共加入氮化钒铁、钒氮合金、钒铁 B、硅铝钙、硅锰面、石油焦增碳剂、锰硅合金、碳化硅、硅钙碳脱氧剂等主要 9 中合金或催化剂，不同合金料的价格不同。根据已知数据，建立合金最小成本模型，通过单纯形法求解各配料投入量最优值，以实现合金化成本最低。

三、模型假设

- 1. 合金元素加入到钢水后不能全部被钢水吸收的部分被灼烧掉。
- 2. 假设在相关性分析中，合金加入量数据极少的不作为影响变量。
- 3. 假设脱氧合金化过程中钢水质量变化只考虑合金加入，不考虑出渣量。

四、符号说明

符号	符号意义
f	合金收得率
Y	钢水质量
ΔY	钢水增加质量
M	连铸正样合金元素含量
N	转炉终点合金元素含量

$\Delta \Sigma$	新炉次和待选炉次碳铝、硅含量的偏差和
$\omega[j]_1$	新炉次 j 元素的初始含量
$\omega[j]_2$	待选炉次 j 元素的初始含量
$\Delta \omega[m_j]$	脱氧合金化过程中元素 j 的变化量
$\Delta \omega[j]$	电极平均增碳量
x_i	第 i 种合金的加入量
$c_{i,j}$	第 i 种合金中 j 元素的含量
ρ	关联度
r	关联系数
Error	预测误差
x	输入值
h_i	隐含层输入值
h_o	隐含层输出值
y_i	输出层输入值
y_o	输出层输出值
d_0	期望输出值
w_{ih}	输入层与中间层的连接权值
w_{ho}	隐含层与输出层的连接权值
b_h	隐含层神经元阈值
b_0	输出层神经元阈值
k	样本数据个数

五、问题一的模型建立与求解

5.1 数据预处理

合金收得率的计算方法题干已给出，为了得到合金收得率，我们对转炉终点 C、Mn 含量和连铸正样 C、Mn 含量的数据进行了分析处理。

第一步：剔除缺失数据——第 252 炉后的转炉终点 Mn 含量数据缺失、第 811 炉后连铸正样中的 C、Mn 含量数据缺失。我们在计算 C 收得率时把第 811 炉后的数据剔除；计算 Mn 收得率时把第 252 炉以后的数据剔除。

第二步：去掉异常数据——转炉终点温度为 0 的明显为异常数据（转炉终点温度为 0 时无法进行脱氧合金化）；转炉终点 C、Mn 含量为 0 的为异常数据（转炉终点 C、Mn 含量不可能为 0）。把这些异常数据剔除。

表 5-1 C 收得率计算异常数据

C 收得率计算异常数据				
总数据	连铸正样 C 数据缺失	转炉终点温度为 0	转炉终点 C 含量为 0	剔除数据总和
1716	906	52	5	963

表 5-2 Mn 收得率计算异常数据

Mn 收得率计算异常数据				
总数据	转炉终点 Mn 数据缺失	转炉终点温度为 0	转炉终点 Mn 含量为 0	剔除数据总和
1716	1465	19	5	1489

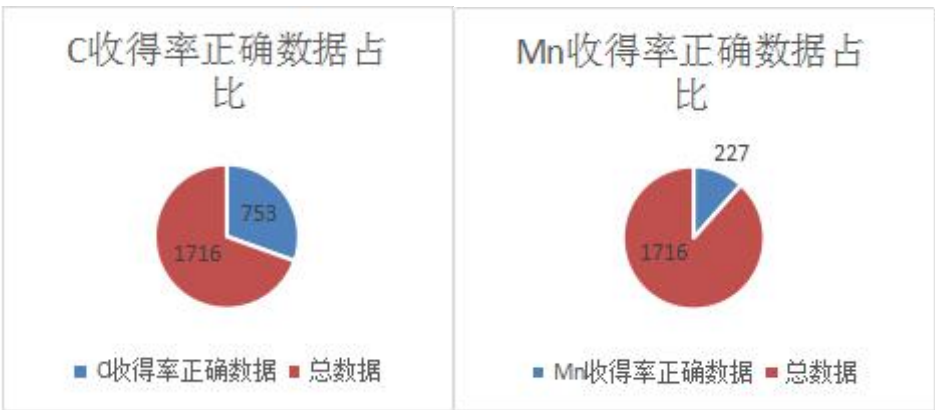


图 5-1 C、Mn 收得率正确数据占比

第三步：数据分类——不同钢种（普碳、低合金）、不同钢号（九种钢号）所对应的合金加入种类不同，把不同钢种不同钢号对应的合金加入种类进行分类，为相关性分析做准备。

5.2 C、Mn 收得率计算模型

根据题干中给出的计算方法：合金收得率指脱氧合金化时被钢水吸收的合金元素的重量与加入该元素总重量之比。我们得到了一般的合金收得率计算模型：

$$f = \frac{(Y + \Delta Y)M - YN}{\sum (x_i \bullet c_{i,j})}$$

(5-1)

其中钢水的质量变化我们将其理想化，脱氧合金化过程中的出渣量予以忽略，钢水的质量变化只考虑合金加入量。

通过公式（5-1）和预处理后的数据分别计算出 C、Mn 的历史收得率，部分数据如下所示（因数据较多，完整数据在附录中展示）：

表 5-3 前十五炉的 C、Mn 收得率

炉次	C 的收得率	Mn 的收得率
1	0.91340509	0.883637978
2	0.866518082	0.920269313
3	1.012027936	0.967153095
4	0.86722293	0.9096801
5	0.97958534	0.932447029
6	0.970512157	0.913351683
7	1.029550787	0.906104418
8	0.915880403	0.910469464
9	0.692779559	0.892347873
10	0.998492657	0.879989312
11	0.973602429	0.859754809
12	0.91488857	0.758034933
13	1.282473144	0.876446964
14	0.827665501	0.756319395
15	0.846969348	0.839212112

观察表 5-1 可以看出 C 的收得率有少部分大于 1（完整数据中约有六分之一的数据大于 1），而 Mn 的收得率全部在 1 以下。为什么会出现 C 收得率大于 1 的情况呢？通过查阅资料^[1]我们发现 C 的收得率大于 1 可能是由于脱氧合金化过程中的电极加热增碳

和钢水以及渣中的氧量有关，因此在考虑电极增碳与氧量的影响下，我们对 C 的收得率计算模型进行了优化。

5.3 基于参考炉次法的自学习优化模型（针对 C 元素收得率）

5.3.1 C 收得率影响因素

（1）电极增碳量

查阅资料^[1]发现，在脱氧合金化过程中与 C 元素有关的因素可能是因为电极加热时增碳。而影响电极增碳的因素也有许多，因素太过复杂不易计算，因此我们理想化影响电极增碳因素相同情况下，计算出电极加热一小时吨钢平均增碳量为： $0.0291\%/(t \bullet h)$ 。

（2）钢水和渣中氧量

在脱氧合金化过程中，可能会脱去大量的氧，因此可能会影响钢水中量，从而影响合金的收得率，但是钢水中脱去氧量不易计算，我们考虑用钢水中的碳、硅和铝含量侧面反应钢水脱氧量。

5.3.2 模型建立

由于影响碳合金收得率的影响因素过多，因此我们采用基于自学习的参考炉次法反算碳元素收得率。

参考炉次法的基本思想为^[1]：在预测新炉次的数据时，我们选取除预测数据外其他数据相近的几个炉次作为参考，利用这几个炉次的收得率平均值作为新炉次的收得率。在预测完新炉次的数据后，对预测的数据进行判断，若数据在标准内，则作为正常炉次，继续预测下一个数据；若数据不在标准内，则不予考虑，以此完成模型的自学习。

在选取参考炉次方面，因为我们要考虑到钢水和渣中氧量影响，所以需要下式进一步选择参考炉次：

$$\Delta \Sigma = \Delta \omega[C] + \Delta \omega[Al] + \Delta \omega[Si] \quad (5-2)$$

$$\Delta \omega[C] = |\omega[C]_1 - \omega[C]_2| \quad (5-3)$$

$$\Delta \omega[Al] = |\omega[Al]_1 - \omega[Al]_2| \quad (5-4)$$

$$\Delta \omega[Si] = |\omega[Si]_1 - \omega[Si]_2| \quad (5-5)$$

我们选择生产条件相近的五个炉次作为参考。

其中合金收得率计算公式为：

$$f_j = \frac{(\Delta\omega[m_j] - \Delta\omega[j]) \cdot (Y + \sum x_i)}{\sum (x_i \cdot c_{i,j})} \times 100\% \tag{5-6}$$

式中， $\Delta\omega[j]$ 为电极平均增碳量。

5.3.3 优化模型求解 C 收得率

通过上述优化模型，我们对 C 的历史收得率进行了再次求解，发现 C 的收得率中只有极个别数据大于 1，通过观察 C 收得率大于 1 的其他数据，发现这可能是由于部分数据异常（例如钢水重量过大、数据残缺等导致的）。得到 C 的部分收得率如下表所示（完整数据在附录中展示）：

表 5-4 前十五炉优化后 C 收得率

炉次	C 的收得率
1	0.81340509
2	0.766518082
3	0.912027936
4	0.76722293
5	0.87958534
6	0.870512157
7	0.929550787
8	0.815880403
9	0.592779559
10	0.898492657
11	0.873602429
12	0.81488857
13	1.182473144
14	0.727665501
15	0.746969348

表 5-2 与表 5-1 相比，可以发现 C 收得率大于 1 的数据明显减少。总的的数据中 C 收得率大于 1 的约为三十分之一（大约为 20 个），相比于之前的六分之一大大减少，并且观察这二十个数据发现，收得率大于 1 的 C 加入量明显比正常数据低，因此可能是合金加入量的数据异常导致的，因此我们把这二十个异常数据删除。

5.4 灰色关联度分析

为分析收得率和其他因素的相关性，由于数据较少，因此我们采用灰色关联度分析来做相关性分析。

5.4.1 数据的选取

通过观察预处理后的数据我们发现合金的加入种类和钢种类型（普碳、低合金）和钢号类型（9种）有关，不同钢号和钢种对应不同合金的相关性应该也不同，因此我们对钢号类型进行了分类（由于普碳钢种只占 0.6%，因此不做分类）：

表 5-5 钢号种类

钢号	20MnKA	20MnKB	HRB400B	HRB400D	HRB500B	HRB500D	Q235	Q235A	Q345B
占比	1.92%	0.75%	75.23%	16.02%	0.99%	3.84%	0.23%	0.40%	0.58%

观察表 5-3 发现 9 个钢种中只有 HRB400B 钢号的数据较多为 75.23%，其他钢号的数据量较少，因此我们只选取 HRB400B 这一钢号来做分析。

HRB400B 这一钢号在脱氧合金化过程中加入的合金种类为八种，分别为：钒氮合金、硅锰面、石油焦增碳剂、锰硅合金 FeMn64Si27、锰硅合金 FeMn68Si18、碳化硅(55%)、硅钙碳脱氧剂。选择这八种合金为灰色关联分析的数据。

另外转炉终点的合金元素含量和转炉终点温度也可能和 C 的收得率有相关性，因此转炉终点的 C、Mn、S、P、Si 含量和转炉终点温度也作为灰色关联分析的数据。

5.4.2 建立灰色关联度分析模型

1. 确定参考序列

以优化后的 C 元素收得率和 Mn 元素收得率为参考序列，其余变量(转炉终点的 C、Mn、S、P、Si 含量、转炉终点温度、八种合金的加入量)为比较序列。

2. 无量纲化处理

我们所采用的变量序列中，有部分数据不统一，所以我们全部把其进行无量纲化，因为序列中有的序列变化量较大，因此采取均值化算子进行无量纲化处理，即：

$$X_i(K) = \frac{X_i(K)}{\frac{1}{n} \sum_{K=1}^n X_i(K)} \quad (5-7)$$

3. 求绝对差值

求出比较序列与参考序列绝对差。

$$\Delta X_0(K) = |X_0(K) - X_0(f)| \quad (5-8)$$

4.求关联系数及关联度

通过比较序列与参考序列的绝对差计算关联系数和关联度。

$$\text{关联系数: } r_i(K) = (\Delta X \min + \theta \Delta X \min) / (\Delta X_0(K) + \theta \Delta X \max) \quad (5-9)$$

$$\text{关联度: } \rho_{0i} = \sum r_i(K) / n \quad (5-10)$$

其中, θ 为分辨系数 (一般是人为给定的, 通常取 0.5, 其值越小分辨率越大)。

5.4.3 模型求解

利用 MATLAB 软件对模型进行求解, 分别得到 C、Mn 收得率和各个影响因素之间的关联度。如下表:

表 5-6 C 收得率关联度

各个参考序列和 C 收得率的关联度						
转炉终点温度	转炉终点 C	转炉终点 Mn	转炉终点 S	转炉终点 P	转炉终点 Si	钒氮合金(进口)
0.858864622	0.80231002	0.654865	0.85457353	0.644815	0.82249611	0.774629475
钒铁 (FeV50-B)	硅铝钙	硅锰面	石油焦增碳剂	锰硅合金	碳化硅	硅钙碳脱氧剂
0.837270414	0.73279066	0.753229226	0.8299126	0.88382030	0.83883996	0.84231288

表 5-7 Mn 收得率关联度

各个参考序列和 Mn 收得率的关联度						
转炉终点温度	转炉终点 C	转炉终点 Mn	转炉终点 S	转炉终点 P	转炉终点 Si	钒氮合金(进口)
0.856646459	0.746582015	0.64649	0.7706365	0.6448558	0.741505899	0.728130738
钒铁 (FeV50-B)	硅铝钙	硅锰面	石油焦增碳剂	锰硅合金	碳化硅	硅钙碳脱氧剂
0.704303383	0.650596212	0.724521391	0.768830292	0.91780554	0.774255017	0.82191773

各个参考序列和 C、Mn 的关联度分别如表 5-6、5-7 所示, 我们分别选取其中关联度较大的因素。

其中, 影响 C 收得率的因素我们选取了七个, 分别为转炉终点温度、转炉终点 C、转炉终点 S、转炉终点 Si、钒铁(FeV50-B)、锰硅合金、碳化硅(55%)。

影响 Mn 收得率的因素我们也选取了七个, 分别为转炉终点温度、转炉终点 C、转炉终点 S、转炉终点 Si、硅铝合金 FeAl30Si25、石油焦增碳剂、锰硅合金。

六、问题二的模型建立与求解

6.1 模型准备

第一问灰色关联度分析我们选取的是 HBR400B 钢号与其他变量做关联性分析，并且选取了七个与 C 收得率相关性较好的因素，分别为：转炉终点温度、转炉终点 C、转炉终点 S、转炉终点 Si、钒铁(FeV50-B)、锰硅合金、碳化硅(55%)来对其进行预测。我们将预处理后的前 620 组数据作为样本，其中 C 的收得率作为期望，并将所有数据作为训练数据确定输入层是 7。将预测的收得率作为输出层层数是 1，根据反复调试确定隐含层节点数，找出最合适的节点数最后计算其误差。

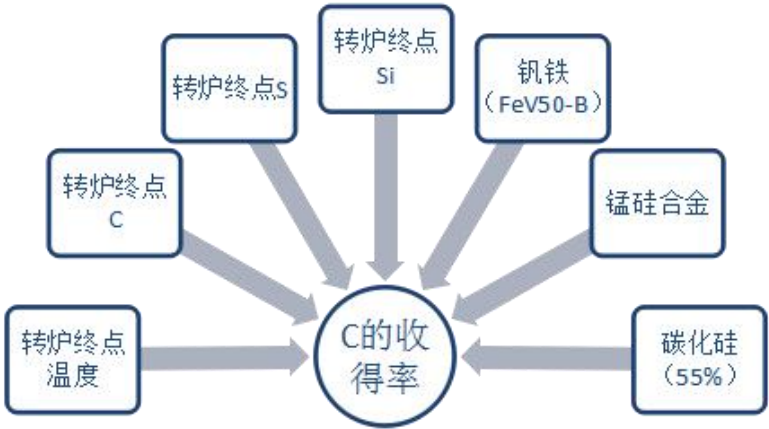


图 6-1 C 的收得率与相关因素

第一问中我们也选取七个与 Mn 收得率相关性较好的因素，分别为：转炉终点温度、转炉终点 C、转炉终点 S、转炉终点 Si、硅铝合金 FeAl30Si25、石油焦增碳剂、锰硅合金来对其进行预测。我们将预处理后的前 248 组数据作为样本，其中 Mn 的收得率作为期望，并将所有数据作为训练数据确定输入层是 7。将预测的收得率作为输出层层数是 1，根据反复调试确定隐含层节点数，找出最合适的节点数最后计算其误差。

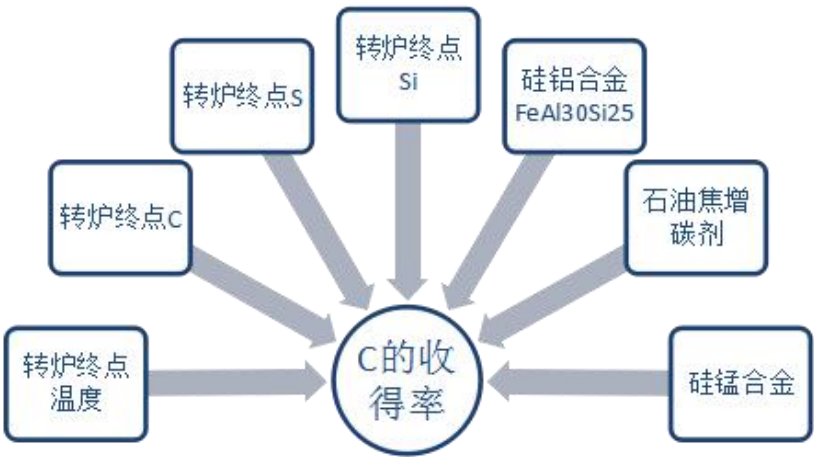


图 6-2 Mn 的收得率与相关因素

6.2 BP 神经网络模型建立

第一步，网络初始化。

我们对 BP 神经网络确定精确度值保证其精确度和最大训练次数使其达到最大训练次数时自动停止运算并且输出。

第二步，选取将前 620 组数据作为样本和对应期望。

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k)) \quad (6-1)$$

$$d_o(k) = (d_1(k), d_2(k), \dots, d_q(k)) \quad (6-2)$$

第三步，通过前面 620 组数据来计算隐含层各神经元的输入和输出

$$hi_h(k) = \sum_{i=1}^n w_{ih} x_i(k) - b_h \quad h = 1, 2, \dots, p \quad (6-3)$$

$$ho_h(k) = f(hi_h(k)) \quad h = 1, 2, \dots, p \quad (6-4)$$

$$yi_o(k) = \sum_{h=1}^p w_{ho} ho_h(k) - b_o \quad o = 1, 2, \dots, q \quad (6-5)$$

$$yo_o(k) = f(yi_o(k)) \quad o = 1, 2, \dots, q \quad (6-6)$$

第四步，通过前 620 组数据网络期望输出和实际输出，用来计算误差函数对输出层的各神经元的偏导数

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{\partial w_{ho}} \quad (6-7)$$

$$\frac{\partial yi_o(k)}{\partial w_{ho}} = \frac{\partial (\sum_h w_{ho} ho_h(k) - b_o)}{\partial w_{ho}} = ho_h(k) \quad (6-8)$$

$$\frac{\partial e}{\partial yi_o} = \frac{\partial (\frac{1}{2} \sum_{o=1}^q (d_o(k) - y_o(k))^2)}{\partial yi_o} = -(d_o(k) - yo_o(k)) yo'_o(k) \quad (6-9)$$

$$= -(d_o(k) - yo_o(k)) f'_i(k) - \delta_o(k) \quad (6-10)$$

第五步，通过前 620 组数据利用输出层各神经元的 $\delta_o(k)$ 和隐含层各神经元的输出来反复修改链接权值 $w_{ho}(k)$ 至合适为止。

$$\Delta w_{ih}(k) = -u \frac{\partial e}{\partial w_{ih}} = -u \frac{\partial e}{\partial w_{ho}} = \mu \delta_o(k) ho_h(k) \quad (6-11)$$

$$w_{ho}^{N+1} = w_{ho}^N + \eta \delta_o(k) ho_h(k) \quad (6-12)$$

第六步，我们通过计算出隐含层各神经元的 $\delta_h(k)$ 和输入层各神经元的值输入反复修改连接权直到合适为止。

$$\Delta w_{ih}(k) = -u \frac{\partial e}{\partial w_{ih}} = -u \frac{\partial e}{\partial h_i(k)} \frac{\partial h_i(k)}{\partial w_{ih}} = \delta_h(k) x_i(k) \quad (6-13)$$

$$w_{ih}^{N+1} = w_{ih}^N + \eta \delta_h(k) x_i(k) \quad (6-14)$$

第七步，计算全局误差

$$error = \frac{1}{2m} \sum_{k=1}^m \sum_{o=1}^q (d_o(k) - y_o(k))^2 \quad (6-15)$$

6.3 模型的求解

利用 MATLAB 对模型进行仿真求解。

由于数据过于庞大这里我们只展示部分 BP 神经网络的拟合和预测结果。

下面为 C 的 BP 神经网络预测

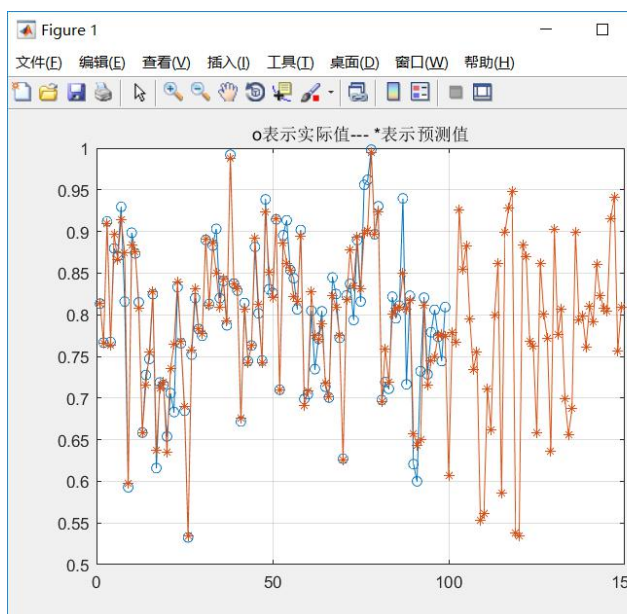


图 6-3 C 的收益率预测图

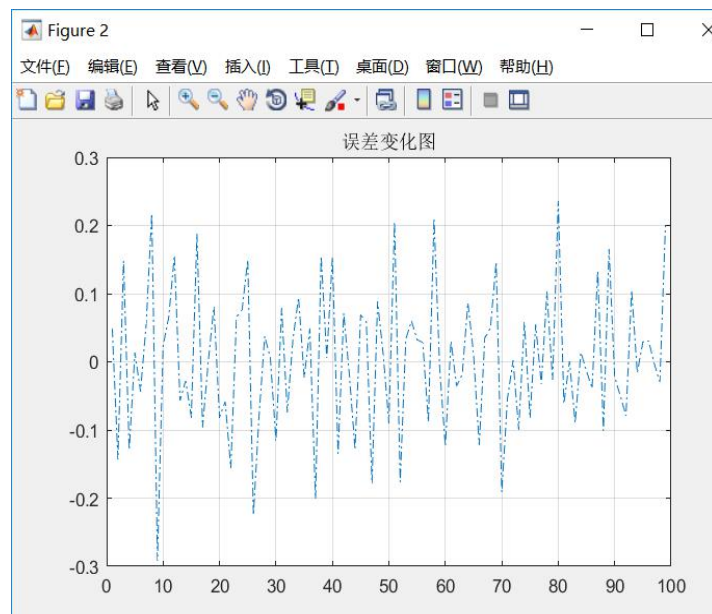


图 6-4 C 的收益率预测误差图

从图 6-3 中可以看出我们 BP 神经网络的预测值的曲线与实际值的曲线拟合十分良好，C 的收益率预测值最高接近于 1，最低在 0.55 左右，大部分在 0.75-0.95 范围内。效果十分良好，在误差分析图 6-4 中，我们看到误差整体在 -0.1 到 0.1 内，少数误差在 -0.2 到 0.2 内，极少数误差可能会在 -0.3 到 0.3 内，误差分析结果较好。

下面为 Mn 的 BP 神经网络预测

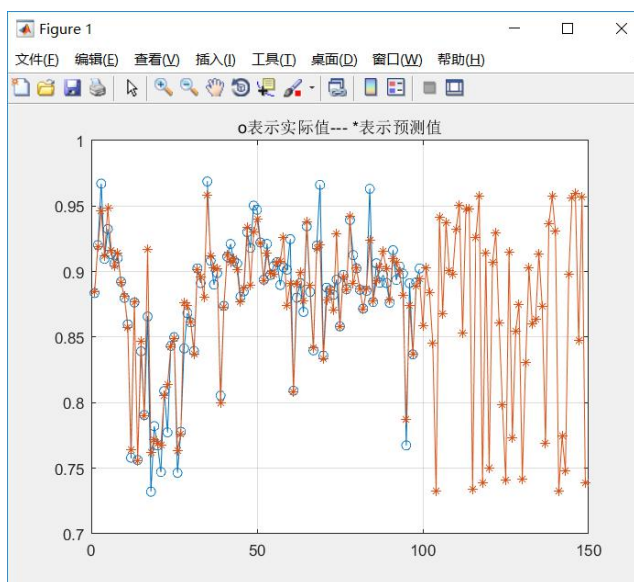


图 6-5 Mn 的收得率预测图

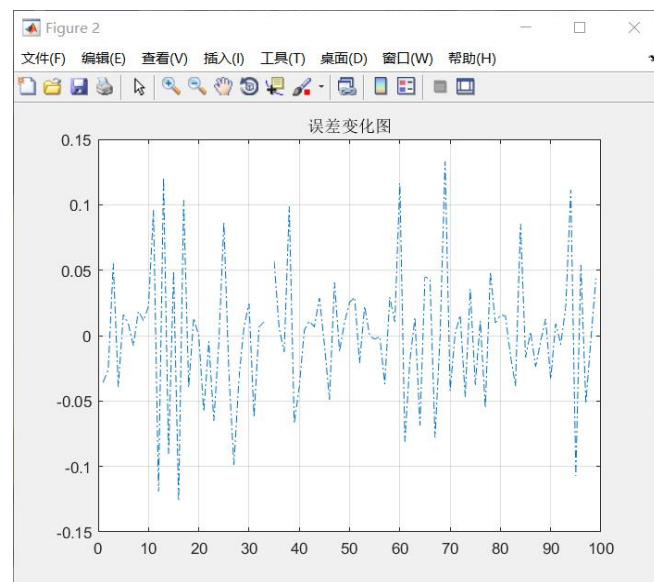


图 6-6 Mn 的收得率预测误差图

从图 6-5 中可以看出我们 BP 神经网络的预测值与实际值十分接近，Mn 的收得率预测值最高在 0.95 左右，最低在 0.75 左右，大部分在 0.85-0.95 范围内。效果十分良好，在误差分析图 6-6 中，我们看到误差整体在 -0.05 到 0.05 内，少数误差在 -0.1 到 0.1 内，极少数误差可能会在 -0.15 到 0.15 内，误差分析结果较好。

6.4 模型的改进

在 BP 神经网络模型中我们的整体数据良好，但是存在一些极少数的较大误差。粒子群算法可以有效的对系统的参数进行优化我们这里选取误差值作为优化目标，适合于实值型处理。十分适合对 BP 神经网络的预测值误差进行优化，这里我们用粒子群算法以预测误差为目标对 BP 神经网络进行了优化。

下面为粒子群算法对 C、Mn 的预测值及误差的优化。

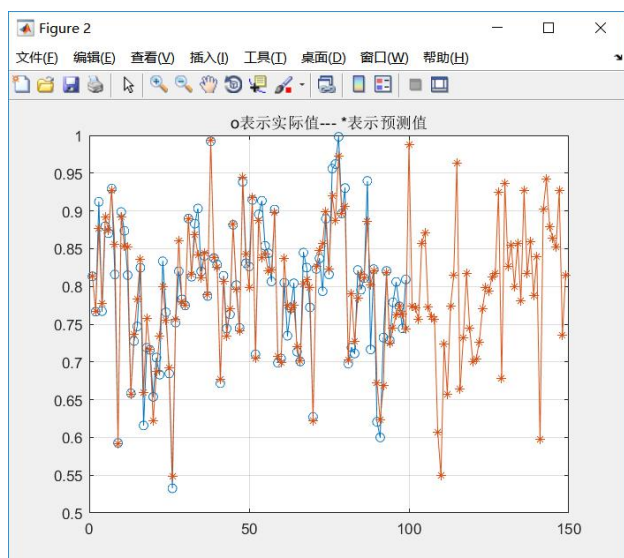


图 6-7 C 的优化后的收得率预测图

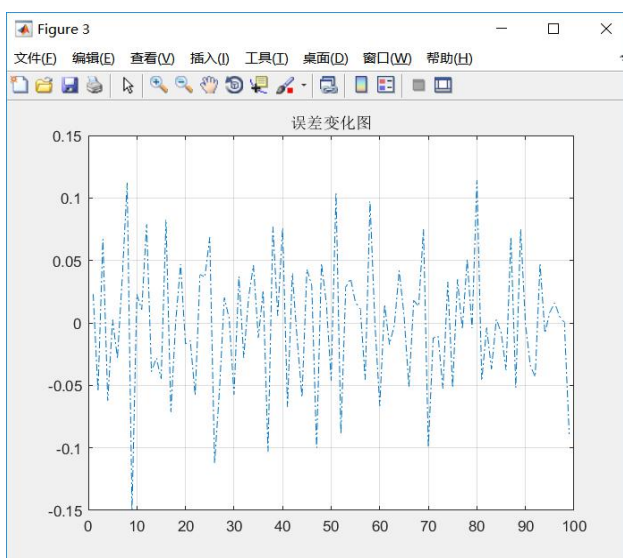


图 6-8 C 的优化后的收得率误差图

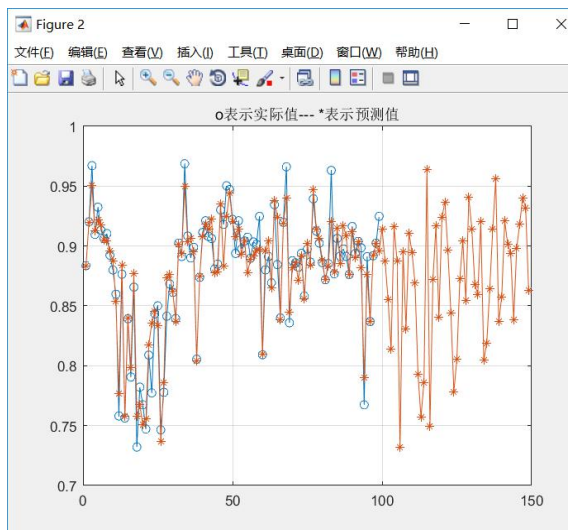


图 6-9 Mn 的优化后的收得率预测图

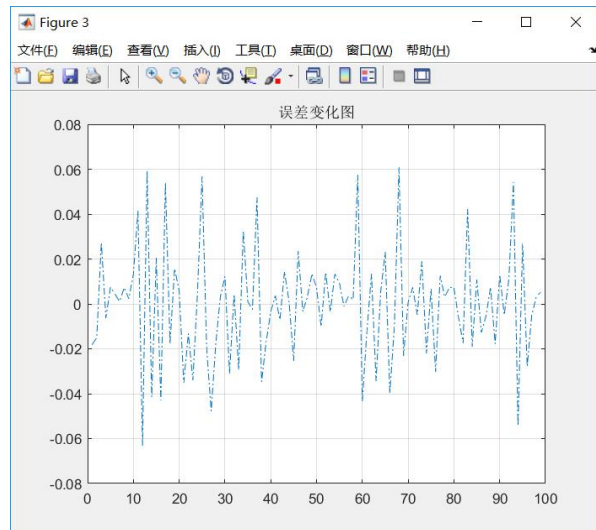


图 6-10 Mn 的优化后的收得率预测误差图

可以发现经过优化后 C、Mn 预测值的误差有较大的改善，其中 C 大部分误差从-0.1 到 0.1 降到-0.04 到 0.04 内，少数误差从-0.2 到 0.2 降到-0.1 到 0.08 内，极少数误差从-0.3 到 0.3 降到-0.15 到 0.15 内。其中 Mn 大部分误差从-0.05 到 0.05 降到-0.02 到 0.02 内，少数误差从-0.1 到 0.1 降到-0.04 到 0.04 内，极少数误差从-0.15 到 0.15 降到-0.08 到 0.08 内。

结果有较大的改善，优化模型的结果十分良好。

七、问题三模型的建立与求解

7.1 模型准备

选取数据量最多，预测效果最好的低合金 HRB400B 进行合金配料成本优化分析。它需要共加入氮化钒铁、钒氮合金、钒铁 B、硅铝钙、硅锰面、石油焦增碳剂、锰硅合金、碳化硅、硅钙碳脱氧剂等主要 9 种合金或催化剂。由于在附件 2 中氮化钒铁、钒氮合金、硅铝钙合金配料里各元素均为 0 或极少，因此只分析余下 6 种合金配料，作为决策变量 x_1, x_2, \dots, x_6 。

在选取合金化元素时，考虑到 5 种合金化元素 C、Mn、S、P、Si 中，S 和 P 在一般情况下是钢中有害元素，硫会使钢产生热脆性；磷会增加钢的冷脆性^[2]。所以在实际脱氧合金化的时候，还会加入除合金配料以外的原料来进行调节硫和磷元素的含量。因此进行成本优化分析时，该两种元素不进行讨论，它们是必须都小于 0.045%。

7.2 模型建立

将各种合金配料的加入量看成线性规划问题。共需合金化的元素为 j ，共 3 种。合金的种类为 n 种，其决策变量、目标函数、约束条件分别为：

1.决策变量

从 7.1 的模型准备可知：每种合金的加入量 x_1, x_2, \dots, x_6 是决策变量。

2.目标函数

以合金成本最低为目标，公式如下：

$$\min Z = a_1x_1 + a_2x_2 + \dots + a_nx_n = \sum_{i=1}^n a_ix_i \quad (7-1)$$

其中， a_i 为第 i 种合金的价格， Z 为所加入合金配料的总成本。

3.约束条件

(1) 对钢水合金元素含量达到国家标准要求

$$Bl_j \leq \frac{\sum_{i=1}^n a_{i,j}f_jx_i + b_jP}{Y + \Delta Y} \leq Bu_j \quad (7-2)$$

$i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m, \quad n = 6, m = 3$, 共 m 个不等式。

式中：

$a_{i,j}$ 为第 i 种合金元素 j 的含量，%；

f_j 为元素 j 的收得率，%；

b_j 为元素 j 在原始钢水中的含量，%；

Y 为原始钢水的重量，kg；

ΔY 为钢水增加的重量，kg；

Bl_j 为钢水中第 j 种元素所需要的下限值，kg；

Bt_j 为钢水中第 j 种元素所需要的上限值，kg。

(2)非负条件

$$x_i \geq 0 \quad i = 1, 2, \dots, 6. \quad (7-3)$$

在上述约束条件中，钢水增加的重量 ΔY 与合金加入量 x_i 有关，说明约束条件不是

线性规划问题的一般形式，所以用改进的单纯形法计算，将约束条件化为线性规划问题的一般形式。

假设合金全部进入钢水中，有

$$\Delta Y = \sum_{i=1}^n x_i \quad (7-4)$$

得到用 x_i 线性表示的约束条件：

$$\begin{aligned} \sum_{i=1}^n (a_{ij} - \frac{Bt_j}{f_j}) x_i &\leq \frac{Y(Bt_j - b_j)}{f_j} \\ \sum_{i=1}^n (a_{ij} - \frac{Bl_j}{f_j}) x_i &\geq \frac{Y(Bl_j - b_j)}{f_j} \end{aligned} \quad (7-5)$$

因此该最小成本模型表示为：

决策变量：每种合金配料的加入量 x_1, x_2, \dots, x_6

$$\text{目标函数：} \min Z = a_1 x_1 + a_2 x_2 + \dots + a_n x_n = \sum_{i=1}^n a_i x_i \quad (7-6)$$

$$\text{约束条件} \begin{cases} \sum_{i=1}^n (a_{ij} - \frac{Bt_j}{f_j}) x_i \leq \frac{Y(Bt_j - b_j)}{f_j} \\ \sum_{i=1}^n (a_{ij} - \frac{Bl_j}{f_j}) x_i \geq \frac{Y(Bl_j - b_j)}{f_j} \\ x_i \geq 0 (i = 1, 2, \dots, n; j = 1, 2, \dots, m) \end{cases} \quad (7-7)$$

可用矩阵表示该模型：

$$\min Z = AX \quad (7-8)$$

$$\begin{cases} BX \leq c \\ X \geq 0 \end{cases} \quad (7-9)$$

这样就构建了一个最小成本的模型，该模型可适合合金中含有多种元素的计算。

7.3 模型求解

选用数据量最多，预测效果最好的低合金 HRB400B 进行合金配料成本优化分析。

将之前分析的 6 中合金配料：钒铁 B、硅锰面、石油焦增碳剂、锰硅合金、碳化硅、硅

钙碳脱氧剂等作为决策变量 x_1, x_2, \dots, x_6 。选取 3 种合金化元素 C、Mn、Si。

线性规划问题的一般描述：

$$\begin{aligned} \min \quad & f^T(x) \\ x \text{ s.t.} \quad & \begin{cases} Ax \leq B \\ A_{eq}x = B_{eq} \\ x_m \leq x \leq x_M \end{cases} \end{aligned} \quad (7-10)$$

求解原理： $[x, f_{opt}, flag, c] = \text{linprog}(f, A, B, A_{eq}, B_{eq}, x_m, x_M, x_0, OPT, p1, p2, \dots)$

将式(7-3)展开为如下约束条件：

$$\text{约束条件} \begin{cases} \sum_{i=1}^n (a_{i1} - \frac{Bt_{j1}}{f_1})x_i \leq \frac{Y(Bt_1 - b_1)}{f_1} \\ -\sum_{i=1}^n (a_{i1} - \frac{Bl_1}{f_1})x_i \leq -\frac{Y(Bl_1 - b_1)}{f_1} \\ \sum_{i=1}^n (a_{i2} - \frac{Bt_2}{f_2})x_i \leq \frac{Y(Bt_2 - b_2)}{f_2} \\ -\sum_{i=1}^n (a_{i2} - \frac{Bl_2}{f_2})x_i \leq -\frac{Y(Bl_1 - b_2)}{f_2} \\ \dots \\ \sum_i^n (a_{im} - \frac{Bt_m}{f_m})x_i \leq \frac{Y(Bt_m - b_m)}{f_m} \\ -\sum_{i=1}^n (a_{im} - \frac{Bl_m}{f_m})x_i \leq -\frac{Y(l_m - b_m)}{f_m} \\ x_i \geq 0 (i = 1, 2, \dots, 6; j = 1, 2, 3) \end{cases} \quad (7-11)$$

将以上不等式组进行系数化简，列写行列式：

$$\text{令 } B_s = \begin{cases} \sum_{i=1}^n (a_{i1} - \frac{Bt_{j1}}{f_1}) \\ -\sum_{i=1}^n (a_{i1} - \frac{Bt_{j1}}{f_1}) \\ \vdots \\ \sum_{i=1}^n a_{im} - \frac{Bt_m}{f_m} \\ -\sum_{i=1}^n a_{im} - \frac{Bt_m}{f_m} \end{cases}, \quad T_s = \begin{cases} \frac{Y(Bt_1 - b_1)}{f_1} \\ -\frac{Y(Bl_1 - b_1)}{f_1} \\ \vdots \\ \frac{Y(Bt_m - b_m)}{f_1} \\ -\frac{Y(Bl_m - b_m)}{f_1} \end{cases}, \quad s = 1, 2, \dots, 2m \quad (7-12)$$

7.4 模型结果

根据处理好以后的附件 1 和附件 2 数据，选取 7A06113 的炉次号，使用 Eviews 的 Genr 函数计算求解矩阵，求得系数的数值如下：

表 7-1 系数 B_s 矩阵的计算值

0.306530669	1.696530669	95.99653067	1.696530669	29.99653067	22.56576147
-0.307363308	-1.697363308	-95.99736331	-1.697363308	-29.99736331	-22.56659411
-0.017807457	66.38219254	-0.017807457	66.38219254	-0.017807457	-0.017807457
0.014468559	-66.38553144	0.014468559	-66.38553144	0.014468559	0.014468559
1.198003686	7.198003686	-0.001996314	17.19800369	55.99800369	39.19800369
-1.184643735	-7.184643735	0.015356265	-17.18464373	-55.98464373	-39.18464373

矩阵 T_s 求解结果如下：

$$T_s^T = [169.75 \quad 111.21 \quad 1165.79 \quad 931.07 \quad -507.38 \quad 431.82]^T$$

使用 Matlab 中的 Linprog() 函数，求解以上不等式，求解得：

Residuals: Primal Dual Upper Duality Total						
Infeas	Infeas	Bounds	Gap	Rel		
A*x-b	A'*y+z-w-f{x}+s-ub	x'*z+s'*w	Error			
Iter 0:	5.01e+04	1.13e+01	4.11e+04	7.12e+05	2.10e+01	
Iter 1:	2.49e+04	2.03e-14	2.04e+04	3.46e+05	1.04e+01	
Iter 2:	2.39e+04	5.80e-12	1.96e+04	1.07e+07	1.00e+01	
x=1.0e+03*(0.0000 0.0134 0.6509 1.5500 0.6364 0.0000)						
f_opt=1.9611e+04						

由 Matlab 求得结果知：x1=0；x2=13.4kg；x3=650.9kg；x4=1550kg；x5=636.4kg；x6=0。最小成本 Z=f_opt=19611 元。即如表格 7-2 所示：

表 7-2 该炉次优化后的合金配料

合金配料	添加量(千克)	价格(元/千克)
钒铁(FeV50-B)	0	205
硅锰面（硅锰渣）	13.4	7.6
石油焦增碳剂	650.9	4.6
锰硅合金 FeMn68Si18	1550	8.15
碳化硅(55%)	636.4	6.1

硅钙碳脱氧剂	0	4
--------	---	---

从表中可知，最小成本为 Z=19611 元。

而该炉次的实际用量为：

表 7-3 该炉次实际的合金配料

合金配料	添加量(千克)	价格(元/千克)
钒铁(FeV50-B)	40	205
硅锰面（硅锰渣）	0	7.6
石油焦增碳剂	85	4.6
锰硅合金 FeMn68Si18	1600	8.15
碳化硅(55%)	132	6.1
硅钙碳脱氧剂	0	4

即该炉次实际成本为 22436.2 元。

因此，计算得实际减少 $w = \frac{22436.2 - 19611}{22436.2} \times 100\% = 12.59\%$

由于大部分钢号的数据缺失，我们针对钢号为 HRB400B 的钢产品再次随机挑选炉次，进行分析，发现算得平均减少成本 10.32%。因此经过我们优化后的合金配料方案，既能使 C、Mn、S、P、Si 等元素达到国家标准，还能达到降低成本防止浪费的目的。可行的合金配料的方案是保持氮化钒铁石油焦增碳剂等合金配料的投入量不变，适当降低锰硅合金 FeMn68Si18 的使用量，大约 10%。碳化硅的主要使用量有 132Kg 和 88kg 两种，与优化结果进行比较并保证效果的前提下，建议碳化硅投入量为 88kg 为宜。

八、问题四 致领导的一封信

致 XX 炼钢厂领导的一封信

尊敬的领导：

您好！

我们是大三的学生，很抱歉在这里打扰您。听闻您所就职的炼钢厂生产出的钢种质量十分优良，并且在国内广受欢迎，我们对您的炼钢厂也十分看好。最近几天我们对您所就职的炼钢厂中部分钢种在脱氧合金化过程中的炼钢数据进行了分析，发现此过程中不同钢号的合金加入量及种类对钢种的质量以及成本方面有一些可以改善的地方。根据我们的分析成果，我们对炼钢过程做了一些优化。以下几点是我们的优化建议，应该可以部分降低您厂的炼钢成本，希望可以对您厂有所帮助：

第一，您厂在国内闻名，可是经过我们对您厂脱氧合金化过程中数据的分析，可能是由于炼钢过程中的疏忽，我们发现您所生产的钢中 Si 元素含量大部分批次并没有达到该钢号的国家标准，希望您能重视这一漏洞，加强钢种的质量把关。

第二，在重点关注 C、Mn、S、P、Si 的含量标准下，我们对您厂不同钢号对应的合金加入量进行了优化，经过我们优化后的合金配料方案，既能使 C、Mn、S、P、Si 等元素达到国家标准，还能达到降低成本防止浪费的目的（大约能降低 5%-13% 的成本）。由于大部分钢号的数据缺失，我们针对钢号为 HRB400B 的钢产品合金配料的方案是保持氮化钒铁石油焦增碳剂等合金配料的投入量不变，适当降低锰硅合金 FeMn68Si18 的使用量，大约 10%。碳化硅的使用量有 132Kg 和 88kg 两种，经分析，建议碳化硅投入量为 88kg 为宜。

希望您能考虑一下我们的建议，同时祝福您的炼钢厂能够越办越好！

此致

敬礼！

2019.4.14

九、模型优缺点

9.1 模型优点

1.考虑到了电极生碳对 C 收得率计算的影响,并且建立基于参考炉次的自学习优化模型来对 C 的收得率进行优化计算,得到较为准确的收得率。

2.BP (神经网络):梯度搜索,细化能力强,可以进行仔细的搜索,可以很好的解决本文非线性问题。

3.粒子群算法粒子群优化具有相当快的逼近最优解的速度,可以有效的对系统的参数进行优化,适合于实值型处理,我们利用粒子群算法优化 BP 神经网络,使预测的误差值明显降低。

9.2 模型缺点

1.最小成本模型核心是单纯形法,这一方法是在元素的目标含量某个上下范围内的计算求解,该方法对成分范围要求较大的钢种很有效。如果目标成分范围要求较小该方法会出现无解的现象。

2.在 C 的收得率优化模型中我们假设钢水质量变化只受合金加入量的影响,忽略了钢渣的质量影响,可能会造成一定的误差。

十、参考文献

- [1] 胡井涛. LF 精炼脱氧合金化模型开发与在线应用[D]. 东北大学, 2011.
- [2] 包燕平, 张超杰, 王敏. 炼钢过程中合金减量化研究现状及展望[J]. 工程科学学报, 2018, 40(09):1017-1026.
- [3] 李维华. 80 号碳化硅脱氧合金化工艺生产实践[J]. 中国冶金, 2018, 28(09):50-53.
- [4] 龚伟, 姜周华, 郑万, 陈念铀. 转炉冶炼过程中合金成分控制模型[J]. 东北大学学报, 2002(12):1155-1157.
- [5] 李朝阳, 李俊国, 孙江波. 转炉优化配料与冶炼静态控制模型[J]. 河北联合大学学报(自然科学版), 2014, 36(03):35-40.
- [6] 韩敏, 徐俏, 赵耀, 林东, 杨溪林. 基于收得率预测模型的转炉炼钢合金加入量计算[J]. 炼钢, 2010, 26(01):44-47.
- [7] 陈为本, 付中华, 吴令. RH 温度预报模型建模方法研究[J]. 工业加热, 2017, 46(06):32-34.

附录

附录 1 问题一 Matlab 相关代码

c 的相关性计算

```
clear;clc;

yangben=xlsread('C:\Users\Administrator\Desktop\C 的收得率计  
算表',1,'S2:S591');

fangzhen=xlsread('C:\Users\Administrator\Desktop\C 的收得率  
计算表',1,'A2:Q591');

[rows,cols]=size(fangzhen);
p=0.5;
[m,n]=size(yangben);
R=[];
for irow=1:rows;
yy=fangzhen(irow,:);
data=[yy;yangben];
data_gyh1=mean(data,2)
for i=1:m+1
for j=1:n
data_gyh(i,j)=data(i,j)/data_gyh1(i,1);
end
end

for i=2:m+1
for j=1:n
Dij(i-1,j)=abs(data_gyh(1,j)-data_gyh(i,j));
end
end
Dijmax=max(max(Dij));
Dijmin=min(min(Dij));

for i=1:m
for j=1:n
Lij(i,j)=(Dijmin+p*Dijmax)/(Dij(i,j)+p*Dijmax);
end
end

LijRowSum=sum(Lij');

for i=1:m
Rij(i)=LijRowSum(i)/n;
```

```

end
R=[R;Rij];
end
R

```

Mn 的相关性计算

```

clear;clc;

yangben=xlsread('C:\Users\Administrator\Desktop\Mn 的收得率
计算表',1,'S2:S200');

fangzhen=xlsread('C:\Users\Administrator\Desktop\Mn 的收得率
计算表',1,'A2:Q200');

[rows,cols]=size(fangzhen);
p=0.5;
[m,n]=size(yangben);
R=[];
for irow=1:rows;
yy=fangzhen(irow,:);
data=[yy;yangben];
data_gyh1=mean(data,2)
for i=1:m+1
for j=1:n
data_gyh(i,j)=data(i,j)/data_gyh1(i,1);
end
end

for i=2:m+1
for j=1:n
Dij(i-1,j)=abs(data_gyh(1,j)-data_gyh(i,j));
end
end
Dijmax=max(max(Dij));
Dijmin=min(min(Dij));

for i=1:m
for j=1:n
Lij(i,j)=(Dijmin+p*Dijmax)/(Dij(i,j)+p*Dijmax);
end
end

```

```

LijRowSum=sum(Lij');

for i=1:m
Rij(i)=LijRowSum(i)/n;
end
R=[R;Rij];
end
R

```

附录 2 问题二 Matlab 相关代码

c 的部分收得率预测

```

clear;clc;

p=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'A2:G100');

%=====期望输出=====

t=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'H2:H100');

ptest=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'A2:G150');

```

```

[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t); %将数据归一化

NodeNum1 =15; % 隐层第一层节点数

NodeNum2=30; % 隐层第二层节点数

TypeNum = 1; % 输出维数
TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
net=newff(minmax(pn),[NodeNum1,NodeNum2,TypeNum],{TF1 TF2
TF3},'traingdx');
%网络创建 traingdm
net.trainParam.show=50;
net.trainParam.epochs=5000; %训练次数设置

net.trainParam.goal=1e-5; %训练所要达到的精度

net.trainParam.lr=0.01; %学习速率
net=train(net,pn,tn);

p2n=tramnmx(p2test,minp,maxp);%测试数据的归一化
an=sim(net,p2n);

[a]=postmnmx(an,mint,maxt) %数据的反归一化，即最终想得到的预测

```

结果

```

z=a(1,2:100);
plot(1:length(t),t,'o-',1:length(t)+50,a,'*-');
title('o 表示实际值--- *表示预测值')
grid on
m=length(a); %向量 a 的长度

t1=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'H2:H100');
error=t1-z; %误差向量

```

```

figure
plot(1:length(error),error,'-.')
title('误差变化图')
grid on

c 全部数据的收得率预测
clear;clc;

p=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'A2:G604')';

%=====期望输出=====

t=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'H2:H604')';

ptest=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'A2:G1262')';

[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t); %将数据归一化

NodeNum1 =15; % 隐层第一层节点数

NodeNum2=30; % 隐层第二层节点数

TypeNum = 1; % 输出维数

TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
net=newff(minmax(pn),[NodeNum1,NodeNum2,TypeNum],{TF1 TF2 TF3},'traingdx');

%网络创建 traingdm

net.trainParam.show=50;

net.trainParam.epochs=5000; %训练次数设置

net.trainParam.goal=1e-5; %训练所要达到的精度

net.trainParam.lr=0.01; %学习速率

net=train(net,pn,tn);

```

```
p2n=tramnmx(p2test,minp,maxp);%测试数据的归一化
an=sim(net,p2n);
```

```
[a]=postmnmx(an,mint,maxt) %数据的反归一化，即最终想得到的预测
```

结果

```
z=a(1,2:604);
grid on
m=length(a); %向量 a 的长度

t1=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'H2:H604');
error=t1-z; %误差向量
plot(1:length(error),error,'-.')
title('误差变化图')
grid on
```

Mn 的部分收得率预测

```
clear;clc;

p=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'A2:H100');
%=====期望输出=====

t=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'I2:I100');
ptest=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'A2:H150');

[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t); %将数据归一化

NodeNum1 =15; % 隐层第一层节点数

NodeNum2=30; % 隐层第二层节点数
```

```

TypeNum = 1;    % 输出维数
TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
net=newff(minmax(pn), [NodeNum1,NodeNum2,TypeNum], {TF1 TF2
TF3}, 'traingdx');

%网络创建 traingdm
net.trainParam.show=50;

net.trainParam.epochs=5000; %训练次数设置

net.trainParam.goal=1e-5; %训练所要达到的精度

net.trainParam.lr=0.01;    %学习速率
net=train(net,pn,tn);

p2n=tramnmx(ptest,minp,maxp); %测试数据的归一化
an=sim(net,p2n);

[a]=postmnmx(an,mint,maxt)    %数据的反归一化，即最终想得到的预测

```

结果

```

z=a(1,2:100);
plot(1:length(t),t,'o-',1:length(t)+50,a,'*-');
title('o 表示实际值--- *表示预测值')
grid on

m=length(a);    %向量 a 的长度

t1=xlsread('C:\Users\Administrator\Desktop\Mn 的数据预
','I2:I100');

error=t1-z;    %误差向量

figure
plot(1:length(error),error,'-.')
title('误差变化图')
grid on

```

Mn 的全部收得率预测

```

clear;clc;

p=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'A2:H248');

%=====期望输出=====

t=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'I2:I248');

ptest=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'A2:H1713');

[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t); %将数据归一化

NodeNum1 =15; % 隐层第一层节点数

NodeNum2=30; % 隐层第二层节点数

TypeNum = 1; % 输出维数

TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
net=newff(minmax(pn), [NodeNum1,NodeNum2,TypeNum], {TF1 TF2 TF3}, 'traingdx');
%网络创建 traingdm
net.trainParam.show=50;
net.trainParam.epochs=5000; %训练次数设置
net.trainParam.goal=1e-5; %训练所要达到的精度
net.trainParam.lr=0.01; %学习速率
net=train(net,pn,tn);

p2n=tramnmx(ptest,minp,maxp); %测试数据的归一化
an=sim(net,p2n);

[a]=postmnmx(an,mint,maxt) %数据的反归一化，即最终想得到的预测

```


结果

```
z=a(1,2:248);  
grid on  
m=length(a);    %向量 a 的长度  
  
t1=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'I2:I248');  
error=t1-z;      %误差向量  
plot(1:length(error),error,'-.'  
title('误差变化图')  
grid on
```

c 的收得率预测改良

```
function main()  
clc;clear all;close all;  
  
tic;                                %程序运行计时  
  
E0=0.001;                          %允许误差  
  
MaxNum=100;                        %粒子最大迭代次数  
  
narvs=1;                          %目标函数的自变量个数  
  
particlesize=30;                   %粒子群规模  
  
c1=2;                              %每个粒子的个体学习因子，也称为加速  
常数  
  
c2=2;                              %每个粒子的社会学习因子，也称为加速  
常数  
  
w=0.6;                             %惯性因子  
  
vmax=0.8;                          %粒子的最大飞翔速度  
  
x=-5+10*rand(particlesize,narvs); %粒子所在的位置  
  
v=2*rand(particlesize,narvs);      %粒子的飞翔速度
```

%用 inline 定义适应度函数以便将子函数文件与主程序文件放在一起,

%目标函数是: $y=1+(2.1*(1-x+2*x.^2)).*exp(-x.^2/2)$

%inline 命令定义适应度函数如下:

```
fitness=inline('1/(1+(2.1*(1-x+2*x.^2)).*exp(-x.^2/2))','x');
```

%inline 定义的适应度函数会使程序运行速度大大降低

```
for i=1:particlesize
    for j=1:narvs
        f(i)=fitness(x(i,j));
    end
end
personalbest_x=x;
personalbest_faval=f;
[globalbest_faval i]=min(personalbest_faval);
globalbest_x=personalbest_x(i,:);
k=1;
while k<=MaxNum
    for i=1:particlesize
        for j=1:narvs
            f(i)=fitness(x(i,j));
        end

        if f(i)<personalbest_faval(i) %判断当前位置是否是历史上
最佳位置

            personalbest_faval(i)=f(i);
            personalbest_x(i,:)=x(i,:);
        end
    end
    [globalbest_faval i]=min(personalbest_faval);
    globalbest_x=personalbest_x(i,:);

    for i=1:particlesize %更新粒子群里每个个体的最新位置

v(i,:)=w*v(i,:)+c1*rand*(personalbest_x(i,:)-x(i,:))...
+c2*rand*(globalbest_x-x(i,:));

        for j=1:narvs %判断粒子的飞翔速度是否超过了最大飞翔速度
            if v(i,j)>vmax;
                v(i,j)=vmax;
            elseif v(i,j)<=-vmax;
                v(i,j)=-vmax;
            end
        end
    end
    k=k+1;
end
```

```

        end
    end
    x(i,:) = x(i,:) + v(i,:);
end
if abs(globalbest_faval) < E0, break, end
k = k + 1;
end
Value1 = 1/globalbest_faval - 1; Value1 = num2str(Value1);
% strcat 指令可以实现字符的组合输出
disp(strcat('the maximum value', '=', Value1));
%输出最大值所在的横坐标位置
Value2 = globalbest_x; Value2 = num2str(Value2);
disp(strcat('the corresponding coordinate', '=', Value2));
x = -2:0.1:3;
y = 2.1 * (1 - x + 2 * x.^2) .* exp(-x.^2/2);
plot(x, y, 'm-', 'linewidth', 3);
hold on;
plot(globalbest_x, 1/globalbest_faval - 1, 'kp', 'linewidth', 4)
;

legend('样本的输出值', '网络的输出值
'); xlabel('X'); ylabel('Y'); grid on; toc;
clear; clc;

p = xlsread('C:\Users\Administrator\Desktop\C 的数据
', 'A2:G100');
%=====期望输出=====

t = xlsread('C:\Users\Administrator\Desktop\C 的数据
', 'H2:H100');
ptest = xlsread('C:\Users\Administrator\Desktop\C 的数据
', 'A2:G150');

[pn, minp, maxp, tn, mint, maxt] = premnmx(p, t); %将数据归一化

NodeNum1 = 15; % 隐层第一层节点数

NodeNum2 = 30; % 隐层第二层节点数

```

```

TypeNum = 1;    % 输出维数
TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
net=newff(minmax(pn), [NodeNum1,NodeNum2,TypeNum], {TF1 TF2
TF3}, 'traingdx');

%网络创建 traingdm
net.trainParam.show=50;

net.trainParam.epochs=5000; %训练次数设置

net.trainParam.goal=1e-5; %训练所要达到的精度

net.trainParam.lr=0.01;    %学习速率
net=train(net,pn,tn);

p2n=trnmnmx(p2test,minp,maxp); %测试数据的归一化
an=sim(net,p2n);

[a]=postmnmx(an,mint,maxt)    %数据的反归一化，即最终想得到的预测

```

结果

```

z=a(1,2:100);
figure
plot(1:length(t),t,'o-',1:length(t)+50,a,'*-');
title('o 表示实际值--- *表示预测值')
grid on

m=length(a);    %向量 a 的长度

t1=xlsread('C:\Users\Administrator\Desktop\C 的数据', 'H2:H100');

error=t1-z;    %误差向量
erroz=0.5*error;
figure
plot(1:length(erroz),erroz,'-.')
title('误差变化图')
grid on

```

Mn 的收得率预测改良

```
function main()
clc;clear all;close all;

tic;                                %程序运行计时

E0=0.001;                           %允许误差

MaxNum=100;                          %粒子最大迭代次数

narvs=1;                             %目标函数的自变量个数

particlesize=30;                     %粒子群规模

c1=2;                                %每个粒子的个体学习因子，也称为加速
常数

c2=2;                                %每个粒子的社会学习因子，也称为加速
常数

w=0.6;                              %惯性因子

vmax=0.8;                            %粒子的最大飞翔速度

x=-5+10*rand(particlesize,narvs);    %粒子所在的位置

v=2*rand(particlesize,narvs);        %粒子的飞翔速度

%用 inline 定义适应度函数以便将子函数文件与主程序文件放在一起，
%目标函数是：  $y=1+(2.1*(1-x+2*x.^2)).*exp(-x.^2/2)$ 

%inline 命令定义适应度函数如下：
fitness=inline('1/(1+(2.1*(1-x+2*x.^2)).*exp(-x.^2/2))','x
');

%inline 定义的适应度函数会使程序运行速度大大降低

for i=1:particlesize
    for j=1:narvs
        f(i)=fitness(x(i,j));
    end
end
personalbest_x=x;
personalbest_faval=f;
```

```

[globalbest_faval i]=min(personalbest_faval);
globalbest_x=personalbest_x(i,:);
k=1;
while k<=MaxNum
    for i=1:particlesize
        for j=1:narvs
            f(i)=fitness(x(i,j));
        end

        if f(i)<personalbest_faval(i) %判断当前位置是否是历史上
最佳位置

            personalbest_faval(i)=f(i);
            personalbest_x(i,:)=x(i,:);
        end
    end
    [globalbest_faval i]=min(personalbest_faval);
    globalbest_x=personalbest_x(i,:);

    for i=1:particlesize %更新粒子群里每个个体的最新位置

v(i,:)=w*v(i,:)+c1*rand*(personalbest_x(i,:)-x(i,:))...
+c2*rand*(globalbest_x-x(i,:));

        for j=1:narvs %判断粒子的飞翔速度是否超过了最大飞翔速度
            if v(i,j)>vmax;
                v(i,j)=vmax;
            elseif v(i,j)<-vmax;
                v(i,j)=-vmax;
            end
        end
        x(i,:)=x(i,:)+v(i,:);
    end
    if abs(globalbest_faval)<E0,break,end
    k=k+1;
end
Value1=1/globalbest_faval-1; Value1=num2str(Value1);

% strcat 指令可以实现字符的组合输出
disp(strcat('the maximum value','=',Value1));

%输出最大值所在的横坐标位置
Value2=globalbest_x; Value2=num2str(Value2);
disp(strcat('the corresponding coordinate','=',Value2));
x=-2:0.1:3;
y=2.1*(1-x+2*x.^2).*exp(-x.^2/2);

```

```

plot(x,y,'m-','linewidth',3);
hold on;
plot(globalbest_x,1/globalbest_faval-1,'kp','linewidth',4)
;

legend('样本的输出值','网络的输出值
');xlabel('X');ylabel('Y');grid on;toc;
clear;clc;
clear;clc;

p=xlsread('C:\Users\Administrator\Desktop\Mn 的数据
','A2:H100');

%=====期望输出=====

t=xlsread('C:\Users\Administrator\Desktop\Mn 的数据
','I2:I100');

ptest=xlsread('C:\Users\Administrator\Desktop\Mn 的数据
','A2:H150');

[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t); %将数据归一化

NodeNum1 =15; % 隐层第一层节点数

NodeNum2=30; % 隐层第二层节点数

TypeNum = 1; % 输出维数
TF1 = 'tansig';
TF2 = 'tansig';
TF3 = 'tansig';
net=newff(minmax(pn),[NodeNum1,NodeNum2,TypeNum],[TF1 TF2
TF3],'traingdx');

%网络创建 traingdm
net.trainParam.show=50;
net.trainParam.epochs=5000; %训练次数设置
net.trainParam.goal=1e-5; %训练所要达到的精度
net.trainParam.lr=0.01; %学习速率
net=train(net,pn,tn);

```

```
p2n=tramnmx(ptest,minp,maxp);%测试数据的归一化
an=sim(net,p2n);
```

```
[a]=postmnmx(an,mint,maxt) %数据的反归一化，即最终想得到的预测
```

结果

```
z=a(1,2:100);
figure
plot(1:length(t),t,'o-',1:length(t)+50,a,'*-');
title('o 表示实际值--- *表示预测值')
grid on
m=length(a); %向量 a 的长度

t1=xlsread('C:\Users\Administrator\Desktop\Mn 的数据', 'I2:I100');
error=t1-z; %误差向量
erroz=0.5*error;
figure
plot(1:length(erroz),erroz,'-.')
title('误差变化图')
grid on
```

附录 3 问题三 Matlab 相关代码

线性规划模型linprog() 求解

```
f=[205 7.6 4.6 8.15 6.1 4]';

A=[0.306530669 1.696530669 95.99653067 1.696530669 29.99653067 22.56576147;
-0.307363308 -1.697363308 -95.99736331 -1.697363308 -29.99736331 -22.56659411;
-0.017807457 66.38219254 -0.017807457 66.38219254 -0.017807457 -0.017807457;
0.014468559 -66.38553144 0.014468559 -66.38553144 0.014468559 0.014468559;
1.198003686 7.198003686 -0.001996314 17.19800369 55.99800369 39.19800369;
-1.184643735 -7.184643735 0.015356265 -17.18464373 -55.98464373 -39.18464373];
```



```
B=[169.7502082 -111.2156536 1165.798553 -931.0740122 -507.3863636 -431.8181818];
```

```
Ae=[]; Be=[];
```

```
xm=[0.001,0,0,1550,0,0.001];
```

```
xM=[0.001,180,85,1550,133,0.001];
```

```
ff=optimset;%ff.LargeScale='off'; % 3»Ê'ÓÃ'ó'æ&ÎÊîâÇó½â
```

```
ff.TolX=1e1; ff.TolFun=1e1; ff.TolCon=1e1; ff.Display='iter';
```

```
[x,f_opt,key,c]=linprog(f,A,B,Ae,Be,xm,xM,[],ff)
```

