

# 电力市场的输电阻塞管理问题

乔磊 常乐 张俊伟

北京工业大学，北京 100022

## 摘要

本文研究了电力市场的输电阻塞管理的优化分配问题。首先利用 MATLAB 多元二项式回归<sup>[1]</sup>对实验数据进行处理，得出各线路上有功潮流与各发电机组出力呈线性关系，也对其合理性进行了检验。得到了然后根据各机组的爬坡速率和下一时段预报的负荷需求作为约束得到各线路上的有功潮流关于各发电机出力的解析关系式，并分析了两种回归的误差范围和适用性。然后根据各机组的爬坡速率和下一时段预报的负荷需求作为约束条件，采用简单的排队法用计算机编程搜索段价表中的最小权值来得到最经济的分配预案。在此基础上讨论了在电网主要线路发生输电阻塞时寻找次优分配方案以及此方案带来的两种阻塞费用的计算方法并采用了其中的一种。次优分配方案的设计转化为一个依据输电阻塞管理原则为约束条件，阻塞费用为目标函数的一组优化问题，运用在分配预案的段价和段容量周围逐步搜索的方法找到使阻塞费用达到最小的次优方案。

运用上面的模型，对问题给出两种预测负荷情况进行预分配和次优分配，通过计算可以得到如下结论：982.4MW 的预测负荷的分配预案超出潮流限值，但是调整后不会出现输电阻塞，1052.8MW 的预测负荷调整后仍然会出现输电阻塞，但不用拉闸限电。

本文采用的模型具有简单、准确、合理等特点。

## 一、问题重述

在电力市场条件下，为保证各电厂的公平竞争，必须按照各电厂的竞价曲线合理分配各电厂的发电功率。但是根据竞价确定的各发电机组出力很可能导致输电阻塞。为了保证系统运行安全，必须消除这种阻塞现象。为此只有使经济效益与阻塞管理相结合，所的结果才具有实际意义<sup>[2]</sup>。

## 二、基本假设

1. 所有的机组出力方案都以保证电网安全的原则，实际运行时不出现一条或几条线路的潮流值的绝对值超过安全裕度。
2. 如果无论如何分配都不能保证线路安全运行时，则采用拉闸限电以强制减少负荷需求，保护电网安全。
3. 每个时段的负荷预报和机组出力分配计划的参照时刻均为该时段结束时刻。机组当前出力是对机组在当前时段结束时刻实际出力的预测值。
4. 发电厂全部机组的所有出力均按当时段的最高段价（即清算价）统一结算。
5. 网方按照分配预案得到的清算价格结算当时段的购电费，当出现分配预案调整时，网方应该为因输电阻塞而不能执行初始交易结果付出代价，网方在结算时应该适当地给发电商以经济补偿。

## 三、符号说明

$L$ : 预测下个时段的负荷要求

$N$ : 主要线路数 ( $N = 6$ )

$T$ : 每个交易时段的时间 ( $T = 0.25h$ )

$N_G$ : 发电机组数量 ( $N_G = 8$ )

$P_{0i}$ : 当前时段各机组的出力 ( $i = 1, 2, \dots, 8$ )

$P_{1i}$ : 下一时段各机组的预分配的出力 ( $i = 1, 2, \dots, 8$ )

$P_{2i}$ : 次优方案分配后各机组的出力 ( $i = 1, 2, \dots, 8$ )

$P_{i\max}$ : 下一时段各机组出力的上限值 ( $i = 1, 2, \dots, 8$ )

$P_{i\min}$ : 下一时段各机组出力的下限值 ( $i = 1, 2, \dots, 8$ )

$R_i$ : 各机组的爬坡速率 ( $i = 1, 2, \dots, 8$ )

$C_{ki}$ : 第  $i$  个机组的第  $k$  段的段容量 ( $i = 1, 2, \dots, 8; k = 1, 2, \dots, 10$ )

$SP_{ki}$ : 第  $i$  个机组的第  $k$  段的段价 ( $i = 1, 2, \dots, 8; k = 1, 2, \dots, 10$ )

$F_{0j}$ : 当前各线路的潮流值 ( $j=1,2,\dots,6$ )

$F_{1j}$ : 按照分配预案计算的潮流值 ( $j=1,2,\dots,6$ )

$F_{2j}$ : 经过调配以后潮流值 ( $j=1,2,\dots,6$ )

$F_j$ : 各线路的潮流限值 ( $j=1,2,\dots,6$ )

$K_j$ : 各线路的相对安全裕度 ( $j=1,2,\dots,6$ )

$M_1$ : 分配预案的清算价

$M_2$ : 次优方案的清算价

$M_3$ : 阻塞费用

#### 四、问题的分析

解决输电阻塞的调度分配问题,最主要的是要了解输电阻塞管理的目标以及市场交易调度中心的具体操作过程和所遵照的原则,以便确定优化对象和约束。

从目标上来看,保证电网安全是第一位的,即不能出现主线路的潮流量超过限值的安全裕度。在安全的前提下,调整分配预案来保证尽量不出现输电阻塞才有意义。

从调度过程来看,调度中心的主要任务是根据下一时段的负荷预报和发电机组的一些资料按照电力市场交易规则得到下一时段的分配预案使达到购电的费用最少的经济目标(即清算价最低),并判断此分配预案是否会引起输电阻塞,如果引起输电阻塞则按照输电阻塞管理原则进行再分配,在解决问题的情况下使阻塞费用最小。这两次分配就是解决输电阻塞管理的核心问题。

从原则来看有二:一是电力市场的交易规则,二是输电阻塞管理原则。电力市场的交易规则的特点是各机组将可用出力由低到高分成 10 段报价,每段有它的段容量和段价,通过由低到高搜索段价来确定每个机组下一时段出力的分配预案,使清算价最低。输电阻塞的管理原则给我们提供了一组消除输电阻塞的方法,根据潮流值的具体情况来选择具体的解决方案和调度方案。

#### 五、数据的处理

为了进一步研究输电阻塞问题,我们围绕当前出力状况的实验数据进行处理,来确定各线路上有功潮流关于各发电机组出力的近似表达式。确定各线路上有功潮流关于各发电机组出力的近似表达式。

令  $x_1, x_2, \dots, x_8$  分别代表各个机组的出力值,  $y_1, y_2, \dots, y_6$  分别代表各条线路的有功潮流值。假设各个机组出力值相互独立,不存在相互关系。且为简单起见,只考虑表达式的形式为多项式形式,即可将  $y_1, y_2, \dots, y_6$  与  $x_1, x_2, \dots, x_8$  的关系分为线性和非线性两种情

况，以下分别讨论：

### 5. 1 线性情况

①假设  $y_j = \beta_{0j} + \beta_{1j}x_1 + \dots + \beta_{8j}x_8$ ,  $i=1,2,\dots,6$  利用多元二项式回归 (linear) 作出图像，并估计出回归系数  $\beta_{ij}$  其中  $i=0,\dots,8$ ;  $j=1,\dots,6$  (所需程序见附录三)

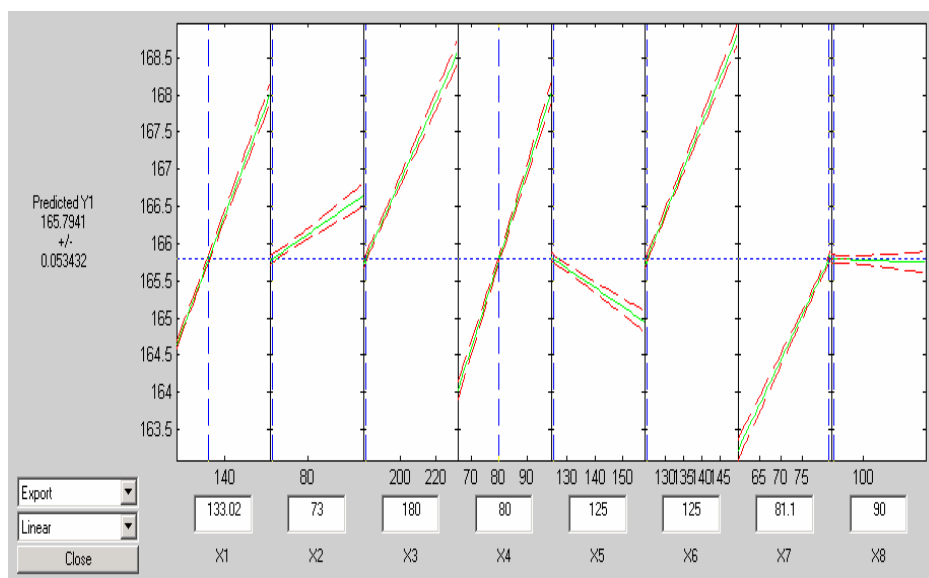


图 1

上图为线路一的潮流值  $y_1$  与机组出力值  $x_1, x_2, \dots, x_8$  的线性关系拟合曲线 (其余图像见附件一 (1))，由此可以看出每个机组出力值  $x_i, i=1, \dots, 8$  与  $y_1$  的拟合曲线均近似为直线，且  $y_1$  的置信区间较小。

②根据所得的回归系数确定的线性关系方程分别为：

$$y_1 = 110.48 + 0.082607x_1 + 0.047764x_2 + 0.052794x_3 + 0.11986x_4 - 0.025705x_5$$

$$+ 0.12165x_6 + 0.12199x_7 - 0.0015179x_8;$$

$$y_2 = 131.35 - 0.054717x_1 + 0.1275x_2 - 0.00014644x_3 + 0.033224x_4 + 0.086667x_5$$

$$- 0.11269x_6 - 0.018644x_7 + 0.098528x_8;$$

$$y_3 = -108.99 - 0.069387x_1 + 0.061985x_2 - 0.1565x_3 - 0.009871x_4 + 0.12467x_5$$

$$+ 0.0023561x_6 - 0.0027873x_7 - 0.20119x_8;$$

$$y_4 = 77.612 - 0.034632x_1 - 0.10278x_2 + 0.20504x_3 - 0.020882x_4 - 0.012018x_5$$

$$+ 0.0056932x_6 + 0.14522x_7 + 0.076336x_8;$$

$$y_5 = 133.13 + 0.00032711x_1 + 0.24283x_2 - 0.06471x_3 - 0.041202x_4 - 0.065452x_5 \\ + 0.070026x_6 - 0.0038961x_7 - 0.0091698x_8;$$

$$y_6 = 120.85 + 0.23757x_1 - 0.060693x_2 - 0.078055x_3 + 0.092897x_4 + 0.046634x_5 \\ - 0.00029128x_6 + 0.16636x_7 + 0.00038828x_8$$

③用以上方程计算出的各线路的潮流值与题目中表 2 的数据极为接近，计算出的潮流值与表 2 中的数据之差较小，即用线性关系近似计算不会有太大的误差。（计算出的潮流值、与表 2 中数据的差值见附件二，所需程序见附件三）

### 5. 2 非线性模型

1. 假设方程形式为纯二次的，即

$$y_i = \beta_{0i} + \beta_{1i}x_1 + \dots + \beta_{8i}x_8 + \beta_{11i}x_1^2 + \dots + \beta_{88i}x_8^2, \quad i = 1, \dots, 6$$

①利用多元二项式回归（purequadratic）作出图像，并估计出回归系数（所需程序见附件三）

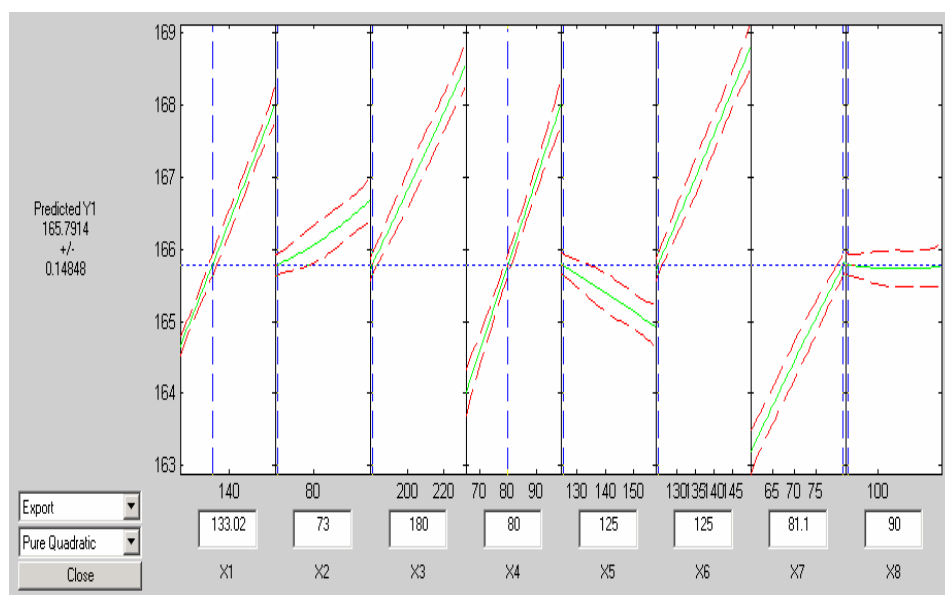


图 2

上图为线路一的潮流值  $y_1$  与机组出力值  $x_1, x_2, \dots, x_8$  的纯二次关系拟合曲线（其余图像见附件一（2）），由图可以看出每个机组  $x_i, i = 1, \dots, 8$  与  $y_1$  的拟合曲线均趋近于直线，二次函数曲线的特征并不明显。

②根据所得的回归系数确定的线路一的潮流值的纯二次关系方程为：

$$y_1 = 119.65 + 0.073046x_1 - 0.10345x_2 + 0.044547x_3 + 0.11905x_4 - 0.020776x_5 \\ + 0.09967x_6 + 0.16751x_7 - 0.041238x_8 + 3.3678e-005x_1^2 + 0.00093514x_2^2$$

$$+1.974e-005 x_3^2 + 4.1084e-00 x_4^2 - 1.9377e-005 x_5^2 + 7.9199e-005 x_6^2 - 0.00030925 x_7^2 + 0.00019138 x_8^2; \text{ (其余方程见附录二 (4))}$$

由以上的方程可以看出二次项的系数与一次项的系数相比非常小而且趋近于 0，在计算的时候与线性方程的效果十分接近。

③从计算出的潮流值来讲（由纯二次方程计算出的潮流值见附录二），确实比线性关系计算出的值更接近与表 2 中的实验结果。（可以通过计算出的潮流值与表二中潮流值差值的大小进行比对）可是用纯二次方程来计算时的计算步骤比用线性方程复杂，但两者的偏差相差极小（由非线性方程计算出的潮流值与表 2 中数据的差值见附录二，所需程序见附录三）。所以我们认为在精度要求不是很高的时候不妨采用线性方程进行近似计算。

2. 多项式阶数未知，用最小二拟合法确定多项式的阶数。

①设方案 0 的出力值为基准值，可以看出表 1 中，把 32 次实验每 4 次实验分为一组，共分 8 组。第 i 组的取值为第 i 台机组的出力值取变化值而其它七台机组的出力值均取基准值。从而也可以把表 2 潮流值分为 8 组，第 i 组的潮流值变化是由第 i 台机组决定的与其他机组无关，便可以把问题转化为一元的最小二乘拟合问题。

②又因为各实验值与基准值相差很小，而且为了方便计算，将变量作如下转化：

$$\begin{aligned} x'_1 &= x_1 - 120; & x'_2 &= x_2 - 73; & x'_3 &= x_3 - 180; & x'_4 &= x_4 - 80; \\ x'_5 &= x_5 - 125; & x'_6 &= x_6 - 125; & x'_7 &= x_7 - 81.1; & x'_8 &= x_8 - 90; \\ y'_1 &= y_1 - 164.78; & y'_2 &= y_2 - 140.87; & y'_3 &= y_3 + 144.25; \\ y'_4 &= y_4 - 119.09; & y'_5 &= y_5 - 135.44; & y'_6 &= y_6 - 157.69; \end{aligned}$$

对新变量进行最小二乘拟合（程序见附录三（7）（8））

③以下为拟合图像：（图中红线代表一次拟合曲线，黑色代表二次拟合曲线，绿色代表三次拟合曲线）

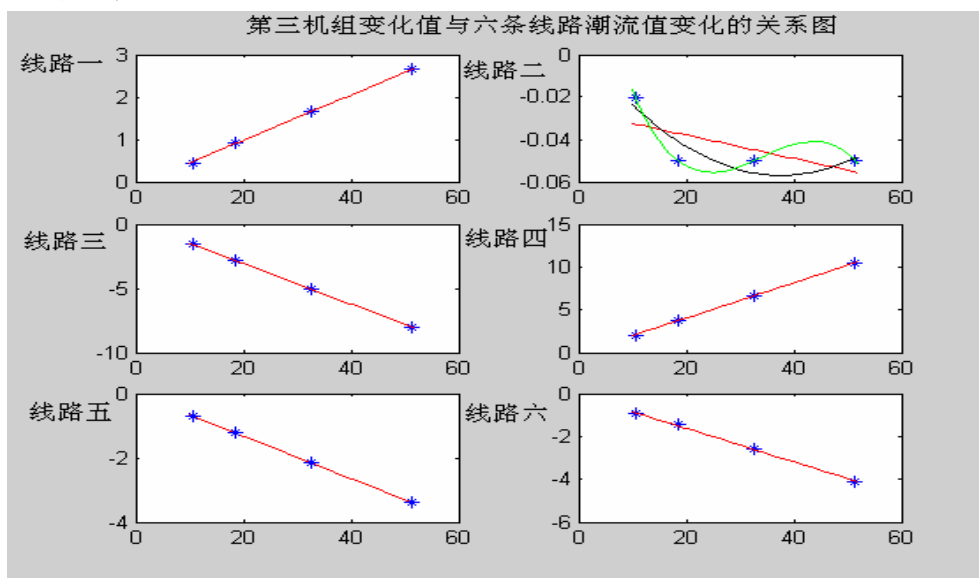


图 3

有以上两图可以看出，大部分机组的出力值与线路的潮流值呈线性关系，个别机组出力值确实与某些线路的潮流值之间存在非线性关系。造成这种非线性关系的原因是：

我们以图 3 中的（线路二）的图为例进行分析：它的纵坐标取值范围只是在 $[-0.06,0]$ 这个小区间内，这只是图 3 中的（线路一）的纵坐标取值范围  $1/50$ ，这相当于将一个拟合曲线用 50 倍的放大镜进行观察，大多数图线都会出现如（线路二）的非线性形式，当（线路二）的纵坐标与（线路一）取相同值时，则也可能会是一条拟合很好的直线。而且图中只有四个点，拟合的时候近似于做其插值，造成曲线呈非线性。

由拟合系数也可看出，非线性关系并不明显；仍以图 3 中的（线路二）为例，在图中我们可看出三次拟合效果很好，可是它的三阶拟合系数分别为

$$b_3=-0.0000 \quad ; \quad b_2=0.0004 \quad ; b_1=0.0135 \quad ; b_0= 0.0800$$

可以看出它的三阶项系数为 0 二阶项系数接近于 0，从而又可证明非线性关系不明显。即可以用线性关系近似计算。

综合（1）（2）可知，在近似计算时可以认为各线路的潮流值与各机组的出力值呈线性关系。

## 六、模型的建立和求解

### 6. 1 下一个时段各机组的出力分配预案

由资料中所给各机组的段价表和段容量表，可以看出如果要取得最少的清算价，应该按照段价表中由低到高的段价所对应的段容量表中该段价对应的容量来分配各机组的出力组合  $P_{li}$ ，直到各机组出力之和等于预测的负荷  $L$ ，此时最后的段价则为最低的

清算价  $M_1$ 。采用购电费用最小为目标函数，近似可表示为： $\min F = \sum_{i=1}^{N_G} P_{li} M_1 T$  此时的约

束条件是由于各机组的当前的出力方案和爬坡速率  $R_i$  而造成的下一时段各机组出力的

上下限值  $P_{i\max}$  和  $P_{i\min}$ ，以及下一时段预报的  $L$ 。约束式如下：

$$\text{s. t. } \sum_{i=1}^{N_G} P_{li} = L$$

$$P_{i\min} \leq P_{li} \leq P_{i\max}, \quad \text{即 } P_{0i} - R_i T \leq P_{li} \leq P_{0i} + R_i T \quad (i=1,2,\dots,N_G)$$

基于以上的目标函数和约束，我们编制 C 语言程序利用计算机搜索段价表中的最小权值来得到最经济的分配预案  $P_{li}$  ( $i=1,2,\dots,N_G$ )。

按照电力市场交易规则，段价从低到高单调不减，每个段价对应一个段容量，按照经济的原则，应按段价由低到高选取选择各个机组的段容量或其部分，直到所有机组的出力之和等于预报的负荷。具体算法如下：

1. 将各机组的段容量和各机组的段价分别构造一个二维数组  $C_{ki}$  和  $SP_{ki}$ ，按照表 3，表 4 初始化，其中  $i$  为机组编号， $k$  为段编号；
2. 设置数组  $\text{current}[i]$ ，记录第  $i$  机组当前出力；初始值为 0；
3. 初始化当前剩余出力，值为下一时段总负荷数；

4. 根据当出力和机组爬坡速率计算出下一时段结束时各机组的出力范围。
5. 在段价数组中搜索最低价格  $SP_{ki}$ ，并作标记（使得下次搜索时不重新搜索此项）记录数组下标  $i, k$ ;
6. 根据当前最低价格所在的数组的下标找到在段容量数组中所对应的段容量  $C_{ki}$ ，判断是取其部分还是全部。即将当前机组  $i$  的出力  $current[i]$  与  $C_{ki}$  相加，如果取值不超过机组  $i$  出力范围，则将  $C_{ki}$  加入到当前机组出力，否则，只取其部分，将这部分加入当前机组出力，并从更新当前剩余出力，即用当前剩余出力值减去新加入的出力作为当前剩余出力；
7. 判断剩余出力值，若果大于零，表明各机组出力之和还没有满足负荷要求，重复执行 5、6 步，否则，算法结束。

此搜索算法执行完毕后的分配方案即在不考虑潮流值的情况下的最优分配方案，也就是初始分配预案。（C 语言程序见附件一）

分别输入两组预测负荷量 982.4MW 和 1052.8MW，分别得到两组分配预案，以及该方案的清算价，问题得解，见下图：

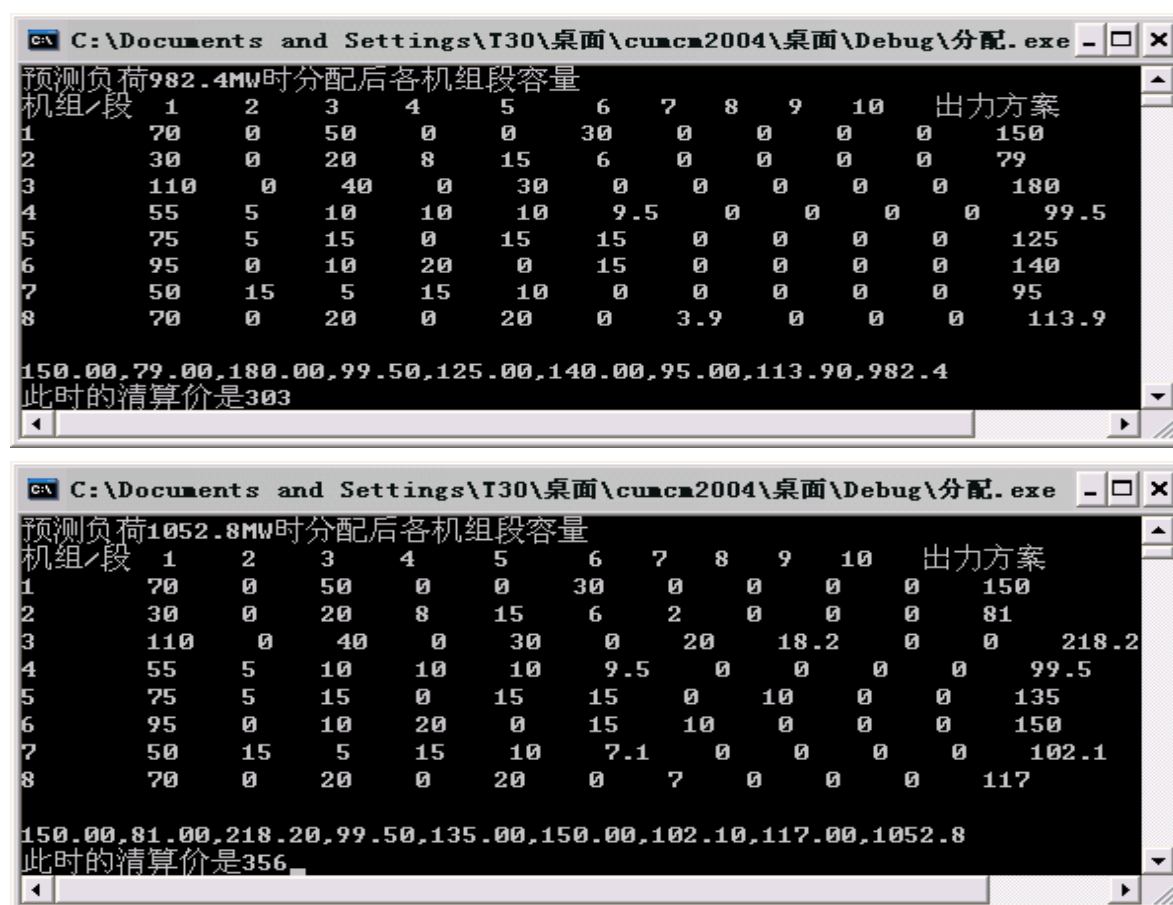


图 4

## 6. 2 阻塞费用的计算规则

当在输电电网中发生阻塞现象时，即当电厂给定的功率传输需要超过了由电网使用者界定的潮流限值时，由于需要在分配预案的基础上重新调整出力方案，因此就出现了阻塞费用。规定序外容量和序内容量均为正数。

第一种可行的计算方案是：当出现传输阻塞时，需要对未纳入发电的计划容量按报价由低到高的顺序安排机组发电，已纳入发电计划的容量按报价由高到低的顺序安排机组停发或少发电。在按照分配预案结算的基础上，多发电量（序外容量）的机组按该机组实际报价由网方支付给发电商，少发电量（序内容量）的机组按清算价和实际报价之差由发电商向网方补偿（即网方在未使用该机组发的电但仍按照实际段价支付该部分电费）。

清算价为  $M_1$ ，次优分配后有  $n_1$  台机组存在序外容量，机组  $i$  存在序外容量为  $\Delta P_i$ ，

对机组  $i$  的序外容量进行单独购买，交易额是当前实际出力的段价  $M_2$ ；有  $n_2$  个机组存在序内容量，机组  $j$  存在序内容量为  $\Delta P_j$ ，实际出力的价段价格为  $M_2'$ ，则交易时对其进行

补偿。阻塞费用  $M_3$  的计算公式为：
$$M_3 = \sum_{i=1}^{n_1} M_2 \Delta P_i - \sum_{j=1}^{n_2} (M_1 - M_2') \Delta P_j$$

还有一种可行的计算方案是：次优分配后有  $n_1$  台机组存在序外容量，机组  $i$  存在序外容量为  $\Delta P_i$ ，在分配预案中出力为  $P_{i1}$ ，因此在高于清算价  $M_1$  上出力，所以网方对机组  $i$  的所有出力功率应该进行单独购买，交易价格是调整后的新的清算价  $M_2$ ；有  $n_2$  个机组存在序内容量，机组  $j$  存在序内容量为  $\Delta P_j$ ，在分配预案中出力为  $P_{j1}$ ，则交易时只对其实际出力购买，交易价格按照分配预案的清算价。由上面所述，阻塞费用  $M_3$  的计

算公式为：
$$M_3 = \sum_{i=1}^{n_1} [M_2 (P_{i1} + \Delta P_i) - M_1 P_{i1}] - \sum_{j=1}^{n_2} M_1 \Delta P_j$$

这两种方法都本着公平的原则，有所不同的是方法一赔付序外容量带来的损失要比方法二要少，但是赔付序内容量的损失要比方法二要多。具体综合来看，用方法一计算得到的阻塞费用更少一些，在公平的基础上，也避免了某些发电企业的投机行为<sup>[2]</sup>。所以后面计算最小阻塞费用时我们选用方法一进行计算。

## 6. 3 各条线路潮流值的计算和输电阻塞的判断

根据 6.1 得到的预测负荷分别为 982.4MW 和 1052.8MW 时的两组分配预案，建起代入 5.1 得到的线性拟合关系式中，计算得到各条线路潮流值如下表：（计算潮流值的 C 语言程序见附件五）

情况\线路	1	2	3	4	5	6
982.4MW	173.307	141.002	-150.92	120.912	136.823	168.521
1052.8MW	177.214	141.165	-156.148	129.744	134.826	167.064

和各线路的潮流限值及安全裕度限值比较,两种情况均发生了输电阻塞,但是各条线路都没有超出安全裕度的限值,因此都需要对分配预案进行再次分配,约束是输电阻塞管理原则,使赔付的阻塞费用最低。

#### 6. 4 根据输电阻塞管理原则和阻塞费用最小进行的次优分配

为了保证电网的安全运行,必须对造成输电阻塞的分配预案进行调整,使电网安全运行,其中包括三方面内容:

(1) 调整各机组出力分配使输电阻塞消除。

(2) 如果无论怎么调配都不能消除输电阻塞,还可以使用线路的安全裕度输电,但是要使每条线路上的潮流的绝对值超过限值的百分比尽量小。

(3) 如果无论怎么调配都无法使每条线路上的潮流的绝对值小于相对安全裕度,则必 987B 拉闸限电。

在此过程中产生了阻塞费用,对于网方来说,分配预案调整的目标是在满足安全用电的情况下使因调整预案造成的阻塞费用最小。因此问题转化成依据输电阻塞管理原则为约束条件,阻塞费用为目标函数的优化问题。对应三方面内容,分别由三套调整方案。

##### 1. 第一步调配方案:

目标函数:

$$\min M_3 = \sum_{i=1}^{n_1} M_2 \Delta P_i - \sum_{j=1}^{n_2} (M_1 - M_2') \Delta P_j = \sum_{i=1}^{n_1} M_2 (P_{2i} - P_{1i}) - \sum_{j=1}^{n_2} (M_1 - M_2') (P_{2j} - P_{1j})$$

约束:  $F_{2j} \leq F_j$

##### 2. 若 1 没有满足条件的 $P_{2i}$ 的解,进行第二步调配方案:

目标函数:  $\min K = \sum_{j=1}^m \frac{F_{2j} - F_j}{F_j}$  ( $m$  为超过限度值的线路的数目,  $m$  应该尽量小)

约束:  $F_{2j} \leq K_j F_j$

##### 3. 若 2 没有满足条件的 $P_{2i}$ 的解,则需要拉闸限电。

由于段价的离散性和调配的随机性,用一个统一的目标函数和一组约束条件实现其来比较困难,因此实际计算过程是在有限范围内对每条阻塞的线路分别调配,使其按全且经济。具体遵循的原则是:

1. 若有多条线路同时发生阻塞,则优先考虑发生阻塞最严重的线路,调整分配,使其消除阻塞,若不能消除,则使其降到超过限值的百分比最小,若还不能使其在安全裕度范围之内,就拉闸限电,这时已经试遍了所有的方案,因此可以不考虑其他线路的情况,直接进行拉闸限电。其次是次严重的线路,以此类推,直到将所有线路

都调整好。

2. 在调整某条线路的阻塞时，观察本线路的潮流与各机组出力的近似表达式，我们称各项前的系数为各机组对此线路潮流值影响的灵敏度，显然，当灵敏度为正时，可降低该机组的出力，当灵敏度为负时，可增加该机组的出力，都能使潮流值减少，从而达到消除阻塞的目的。遵循费用尽可能小的原则，对于灵敏度为负值的机组，按照价格由低到高增加出力，对于灵敏度为正值的机组，按照价格由高到低减少出力。
3. 调整时，最好在清算价及其相邻的段价内调整，这样价格的跨度比较小，符合优化原理和生活常理。
4. 在上述三条原则基础上，对每种组合进行对应的段容量等约束，找出相应的最优解，让后再从所有最优解中找到最优，作为本条线路阻塞费用的最小值，由此看好像是穷举法，但实际操最过程中，有些因素的改变对费用影响极少，因此只需调整几组解之后，便可看到每次的最优解就围绕几个数值作微小的变动，因此只需选这几个解中的最小值即可。

用此方法，调整输电阻塞后的分配方案为：

负荷\机组	1	2	3	4	5	6	7	8	阻塞费用
982.4	120	81	180	95.60988	142	150	96.79002	117	932.2702
1052.8	132.0382	81	228	95.66181	142	155	102.1	117	1279.877

程序见附录三（9）

由此结果我们得到如下结论：982.4MW 的预测负荷的分配预案超出潮流限值，但是调整后不会出现输电阻塞，1052.8MW 的预测负荷调整后仍然会出现输电阻塞，但不用拉闸限电。

## 七、模型的评价

由于非线性模型的比对可以看出线性模型具有简单、准确、使用的特点。利用 R 软件进行显著性检验也可看出模型线性性质明显。

此模型中回归系数 具有实际意义。当  $\beta_i > 0$  时，在消除阻塞过程中第  $i$  台机组应减小出力，相反，当  $\beta_i < 0$  时应增加出力。学术上称  $\beta_i$  为灵敏度系数<sup>[3]</sup>。

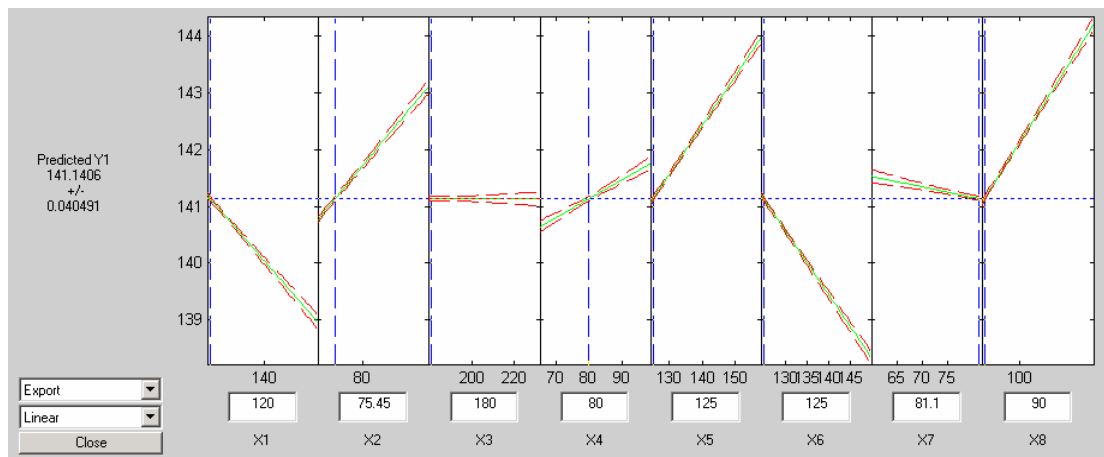
阻塞费用计算时，由于使用的是分步优化的算法，每个解都是局部最优的，但是可能出现的情况是局部最优中的最优不一定是整体中的最优。而且由于条件的限制，不可能将每种组合多算出来，可能导致忽略了一部分解。因此该模型有待改进的地方在于找到一种方法统一约束条件，统一目标函数，直接求出整体的最优解。

## 附录

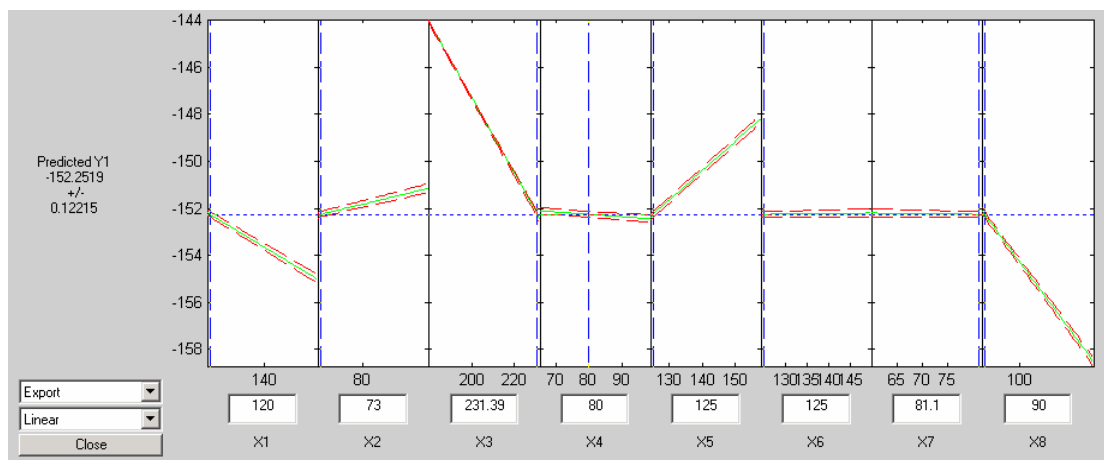
### 附录一：拟合曲线

(1) 各线路潮流值  $y_1, y_2, \dots, y_6$  分别对于每个机组出力  $x_1, x_2, \dots, x_8$  的线性关系拟合图像分别为：

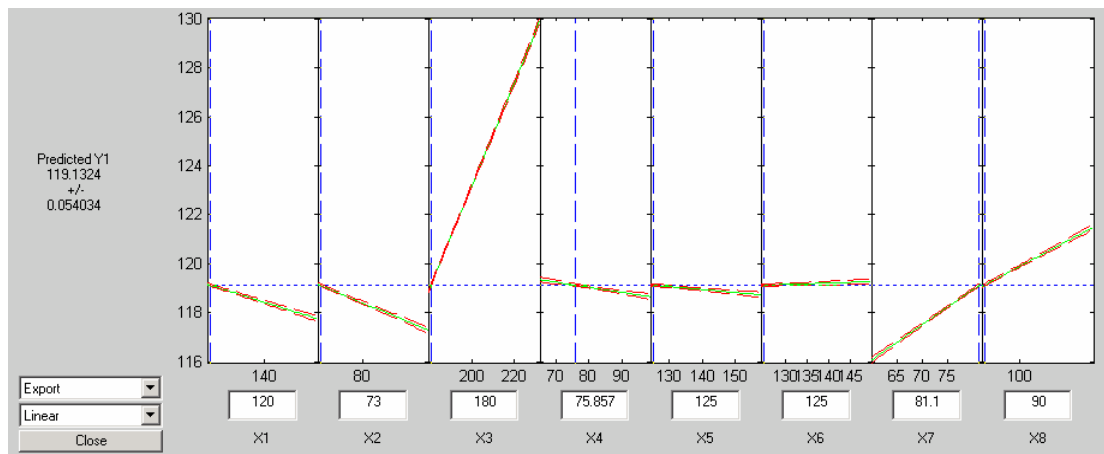
①线路二的潮流值  $y_2$  分别对于  $x_1, x_2, \dots, x_8$  的线性关系拟合曲线为：



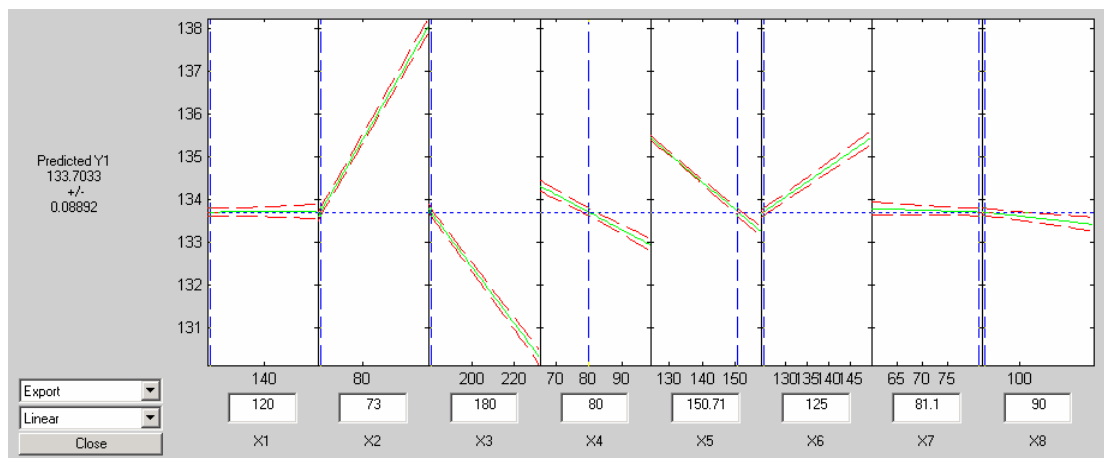
②线路三的潮流值  $y_3$  分别对于  $x_1, x_2, \dots, x_8$  的线性关系拟合曲线为：



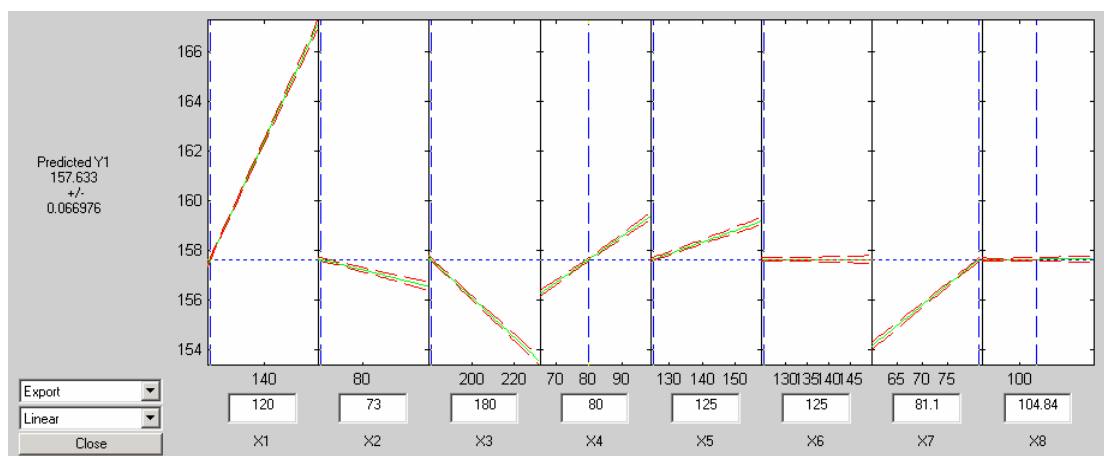
③线路四的潮流值  $y_4$  分别对于  $x_1, x_2, \dots, x_8$  的线性关系拟合曲线为：



④线路五的潮流值  $y_5$  分别对于  $x_1, x_2, \dots, x_8$  的线性关系拟合曲线为:

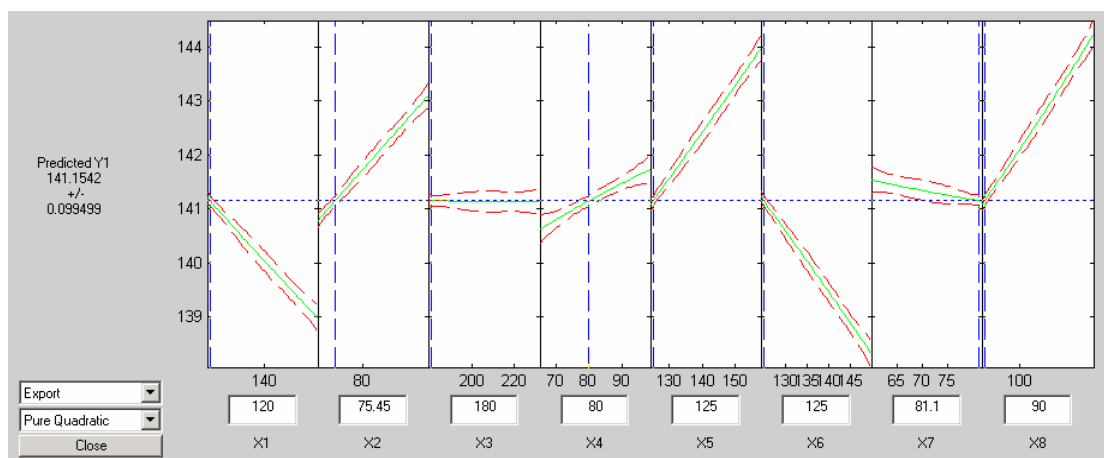


⑤线路六的潮流值  $y_6$  分别对于  $x_1, x_2, \dots, x_8$  的线性关系拟合曲线为:

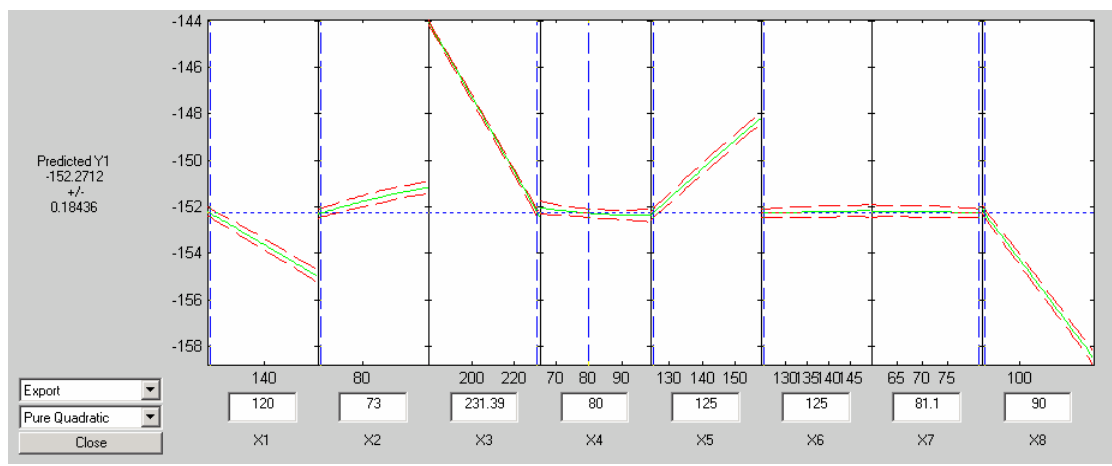


(2) 各线路潮流值  $y_1, y_2, \dots, y_6$  分别对于每个机组出力  $x_1, x_2, \dots, x_8$  的非线性关系拟合图像分别为:

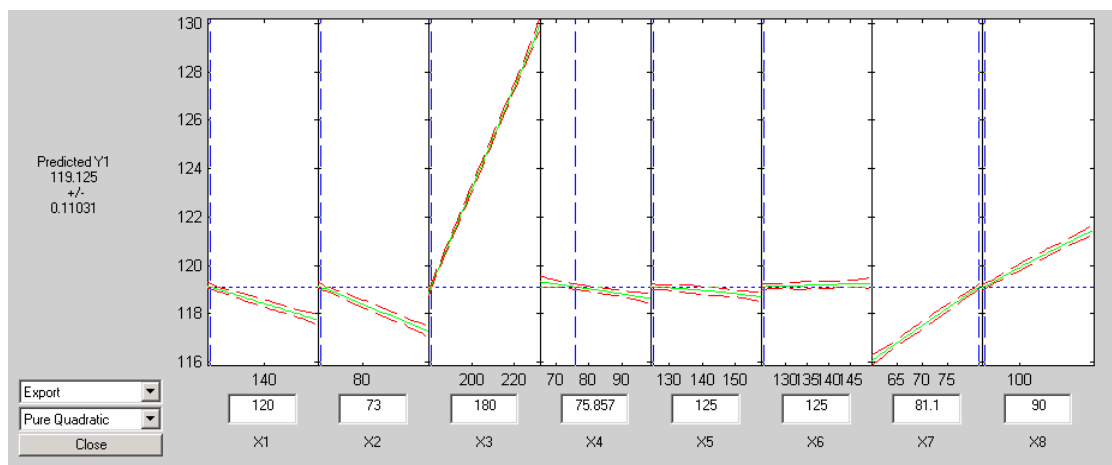
① 线路二的潮流值  $y_2$  分别对于  $x_1, x_2, \dots, x_8$  的非线性关系拟合曲线为:



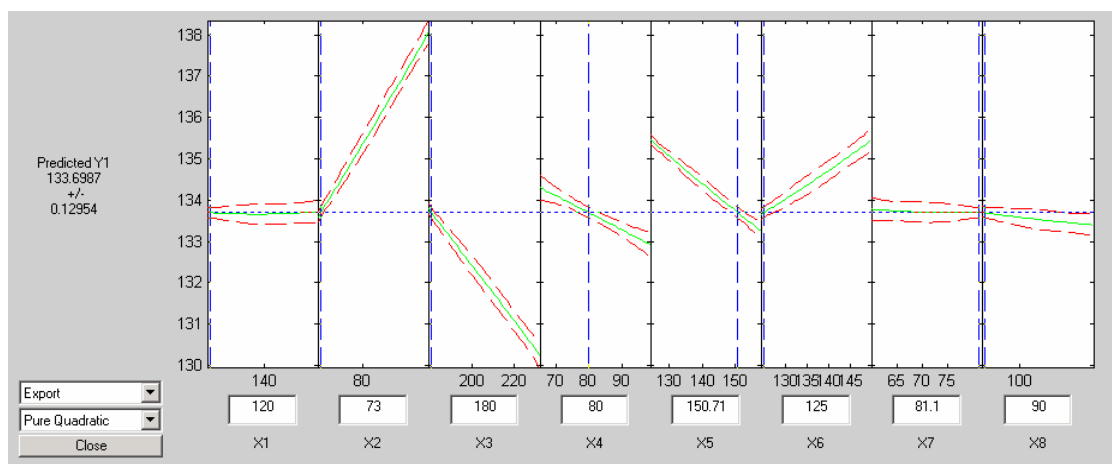
② 线路三的潮流值  $y_3$  分别对于  $x_1, x_2, \dots, x_8$  的非线性关系拟合曲线为:



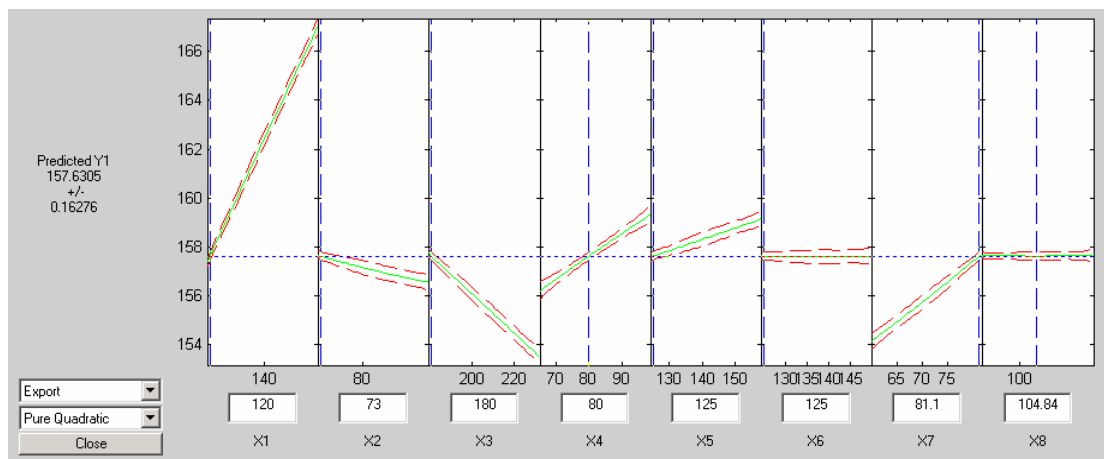
③ 线路四的潮流值  $y_4$  分别对于  $x_1, x_2, \dots, x_8$  的非线性关系拟合曲线为:



④ 线路五的潮流值  $y_5$  分别对于  $x_1, x_2, \dots, x_8$  的非线性关系拟合曲线为:

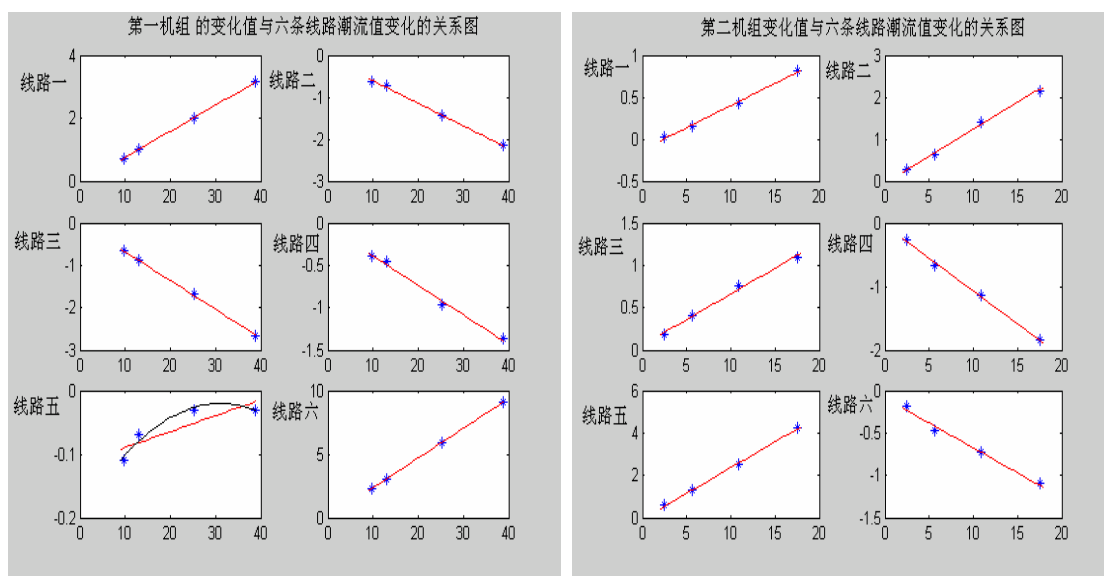


⑤ 线路六的潮流值  $y_6$  分别对于  $x_1, x_2, \dots, x_8$  的非线性关系拟合曲线为:

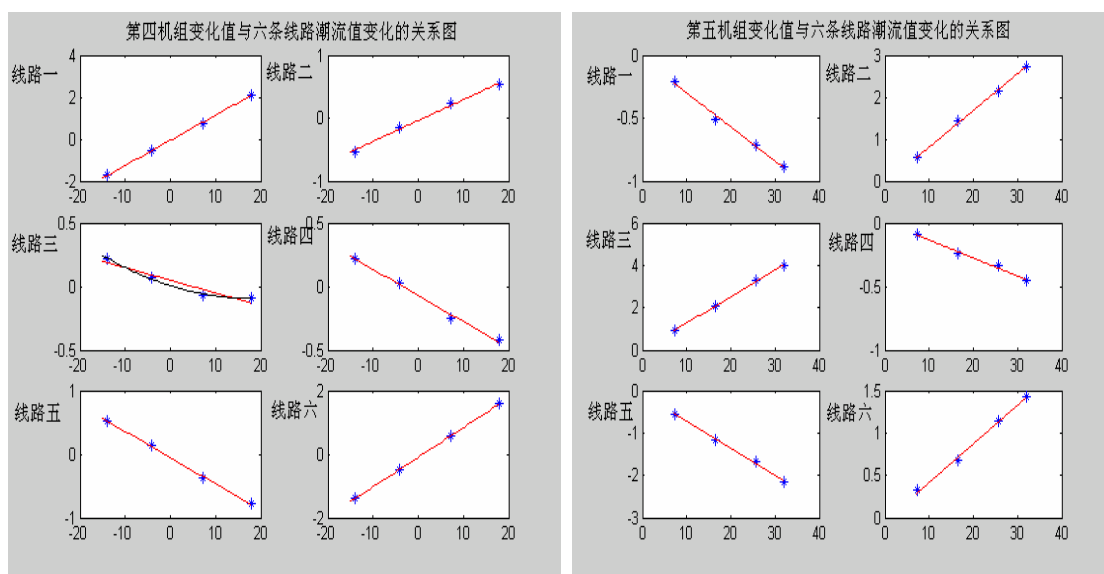


(3) 各台机组变化值与六条线路潮流值变化的关系图如下：

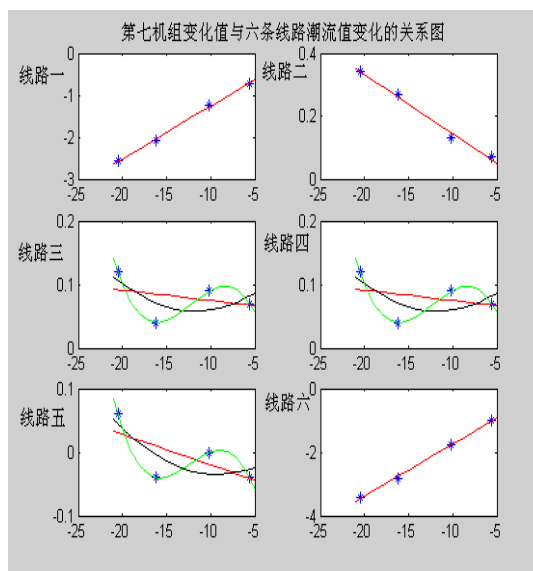
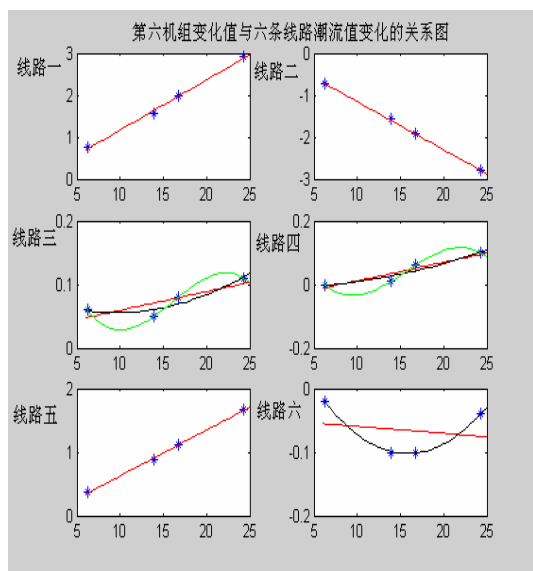
① 第一、二台机组变化值与六条线路潮流值变化的关系图为：



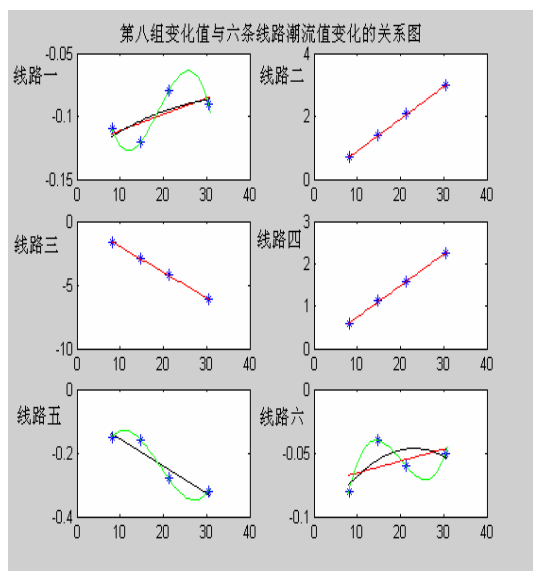
② 第四、五台机组变化值与六条线路潮流值变化的关系图



③ 第六、七台机组变化值与六条线路潮流值变化的关系图



④ 第八台机组变化值与六条线路潮流值变化的关系图



附录二、用拟合出的关系式检验后的数据

(1) 由线性回归方程计算出的潮流值分别为：

164.721	140.826	-144.206	119.047	135.382	157.629
165.797	140.113	-145.11	118.596	135.387	160.722
165.517	140.299	-144.874	118.713	135.386	159.917
167.924	138.704	-146.896	117.704	135.395	166.84
166.813	139.44	-145.963	118.17	135.391	163.645
164.989	141.539	-143.859	118.472	136.741	157.29
164.838	141.138	-144.054	118.795	135.977	157.481
165.556	143.055	-143.122	117.25	139.629	156.568
165.239	142.209	-143.534	117.932	138.017	156.971
167.434	140.818	-152.249	129.584	132.057	153.618
165.697	140.823	-147.098	122.836	134.187	156.187
166.444	140.821	-149.314	125.739	133.27	155.082
165.278	140.824	-145.857	121.21	134.7	156.806

164.225	140.688	-144.165	119.133	135.553	157.244
163.038	140.359	-144.067	119.34	135.961	156.325
165.591	141.067	-144.278	118.895	135.083	158.303
166.858	141.418	-144.382	118.675	134.648	159.285
164.06	143.054	-141.001	118.738	133.7	158.828
164.295	142.263	-142.139	118.848	134.297	158.402
164.532	141.464	-143.287	118.958	134.9	157.973
163.9	143.593	-140.225	118.663	133.292	159.118
166.41	139.262	-144.173	119.126	136.354	157.625
165.477	140.126	-144.191	119.082	135.817	157.627
166.754	138.943	-144.167	119.142	136.553	157.624
167.676	138.088	-144.149	119.185	137.083	157.622
162.218	141.208	-144.149	116.067	135.462	154.216
163.485	141.015	-144.178	117.575	135.422	155.943
162.739	141.129	-144.161	116.688	135.446	154.927
164.042	140.93	-144.191	118.238	135.404	156.702
164.699	142.288	-147.192	120.18	135.246	157.635
164.689	142.916	-148.475	120.667	135.188	157.637
164.709	141.623	-145.834	119.665	135.308	157.632
164.675	143.825	-150.33	121.371	135.103	157.641

(2) 由线性回归方程计算出的潮流值与表 2 中的数据之差值为:

-0.058765	-0.0443626	0.0438975	-0.043158	-0.0576225	-0.0607478
-0.0132219	-0.0167779	0.0304787	-0.0340666	0.0166365	-0.0375864
0.00674041	0.0487127	0.0457007	0.0133358	0.0555276	-0.0629487
-0.00609161	-0.00574069	0.0137635	-0.0158406	-0.0149405	0.0298411
0.0228442	-0.00979704	-0.0429814	0.0399598	-0.0193401	0.0045246
0.0485223	0.0391274	-0.0192345	0.0416851	0.0212542	0.0696142
0.0382568	0.0080124	0.0157607	-0.024969	-0.042689	-0.0194456
-0.0335159	0.0252299	0.0378292	0.00952814	-0.0312543	-0.0220863
0.0293789	-0.0712426	-0.0436892	-0.0281154	0.0365973	0.0108545
0.00431866	-0.00188815	0.0113625	0.0038476	0.0169306	0.0180058
-0.0131319	0.00293119	-0.0182225	-0.0140188	-0.0234633	-0.0432042
-0.00556884	0.000857598	0.0157375	-0.0106524	-0.00975691	-0.008463
0.0482117	-0.0259075	-0.0371775	0.050014	-0.050313	0.0357719
-0.00534498	-0.0420096	0.014793	0.0133561	-0.0169226	0.0443799
-0.00183912	0.019106	-0.0374939	0.030067	-0.00906403	0.0147925
0.0511789	-0.0332228	0.0422538	0.0552804	0.0233334	0.0434986
-0.0223804	0.017822	-0.0420432	0.00464123	-0.022007	0.00504833
-0.00964055	0.023846	-0.0308368	-0.0121408	-0.0503934	-0.00178766
0.0250461	-0.0274237	0.0109261	-0.00241644	0.0271833	0.0324439
-0.0382108	0.0243732	0.0127154	-0.0417307	0.0199963	-0.0370552
0.0104744	-0.0170853	0.0246106	0.0231073	0.0124951	-0.00172418
0.059737	-0.0284998	0.0266001	0.0258636	0.0243384	0.0352092
-0.0633185	-0.0141675	-0.00147115	-0.00780323	0.00723895	-0.0425566

0.0040065	-0.0074125	0.0032679	-0.00802463	0.00251195	0.0343849
-0.0138865	0.0183973	-0.00887286	-0.00487017	-0.026691	-0.027823
0.00824418	-0.00182501	-0.0189127	0.037218	-0.0376823	-0.0441223
-0.0554996	0.0146503	-0.0178449	0.0146016	-0.0181238	0.0126945
0.0393855	-0.0114722	0.0491799	-0.0524021	0.0456735	0.0465676
-0.0183713	-0.0104969	-0.0105745	-0.00217862	0.00408266	0.0224606
0.0387094	0.0177929	0.00823787	-0.0303318	-0.0337023	-0.0149857
-0.0109748	-0.0235984	-0.0253543	-0.0133081	0.0277943	0.0074915
0.0389522	0.062926	0.045868	-0.0154471	0.0181755	0.0223942
-0.0149699	-0.0151703	0.00967387	0.0305098	-0.0167512	0.00107144

(3) 由非线性回归方程计算出的潮流值分别为:

164.727	140.842	-144.249	119.042	135.4	157.643
165.789	140.108	-145.125	118.589	135.366	160.761
165.511	140.297	-144.895	118.706	135.371	159.953
167.923	138.717	-146.9	117.709	135.419	166.804
166.802	139.434	-145.965	118.166	135.371	163.668
164.941	141.555	-143.838	118.473	136.721	157.261
164.813	141.155	-144.062	118.793	135.974	157.47
165.591	143.049	-143.16	117.246	139.65	156.597
165.196	142.218	-143.505	117.935	137.994	156.943
167.433	140.819	-152.275	129.586	132.033	153.606
165.688	140.829	-147.089	122.832	134.213	156.202
166.434	140.823	-149.3	125.737	133.285	155.089
165.274	140.834	-145.864	121.205	134.725	156.821
164.231	140.7	-144.197	119.125	135.57	157.254
163.046	140.343	-144.029	119.318	135.973	156.311
165.596	141.08	-144.314	118.89	135.099	158.315
166.862	141.404	-144.347	118.657	134.655	159.272
164.055	143.054	-141.002	118.743	133.695	158.827
164.296	142.258	-142.137	118.872	134.284	158.401
164.537	141.466	-143.302	118.977	134.897	157.978
163.889	143.603	-140.24	118.641	133.304	159.121
166.4	139.28	-144.185	119.121	136.335	157.613
165.472	140.15	-144.218	119.076	135.805	157.62
166.745	138.954	-144.174	119.139	136.539	157.616
167.675	138.072	-144.148	119.19	137.108	157.639
162.189	141.215	-144.16	116.054	135.476	154.246
163.505	141.014	-144.18	117.587	135.407	155.912
162.739	141.129	-144.162	116.691	135.44	154.919
164.063	140.933	-144.205	118.248	135.397	156.679
164.668	142.265	-147.203	120.205	135.239	157.633
164.669	142.904	-148.48	120.681	135.182	157.635
164.684	141.607	-145.857	119.686	135.307	157.635

164.697      143.856      -150.333      121.341      135.112      157.645

(4) 由非线性回归方程计算出的潮流值与表 2 中的数据的数据的差值为:

-0.0533904 -0.0282343 0.000784339 -0.0484809 -0.0402555 -0.0472151  
 -0.0213854 -0.0217759 0.0151789 -0.0410013 -0.00438001 0.00135531  
 0.00100234 0.0472625 0.0245704 0.00631861 0.0405532 -0.0265315  
 -0.00740811 0.00719845 0.0103141 -0.0106058 0.00897861 -0.0060756  
 0.0123798 -0.0161231 -0.045469 0.0362932 -0.0389181 0.028441  
 0.00101183 0.0547196 0.00238426 0.043245 0.000988902 0.0414714  
 0.0132698 0.0246898 0.00828752 -0.0265655 -0.0463044 -0.029554  
 0.00104853 0.0185377 7.18042e-005 0.00647568 -0.00990763 0.00676027  
 -0.0144874 -0.0619472 -0.0152742 -0.0253669 0.0143245 -0.0165232  
 0.0032098 -0.00103165 -0.0150151 0.00634687 -0.00693613 0.00613217  
 -0.0220942 0.00943025 -0.0089945 -0.0180844 0.00272852 -0.0283537  
 -0.0163931 0.00312267 0.029868 -0.0125724 0.00522207 -0.000547239  
 0.0437501 -0.0158456 -0.0440375 0.0451949 -0.0246543 0.051495  
 0.00073257 -0.0302095 -0.0173726 0.00545159 0.000105466 0.0542929  
 0.00648914 0.00325627 0.000832251 0.0078142 0.00304625 0.00121685  
 0.0556619 -0.019816 0.00635463 0.0496106 0.0388485 0.0545385  
 -0.0184215 0.00381619 -0.00737306 -0.0126164 -0.0147535 -0.00764677  
 -0.0148953 0.0237621 -0.0318242 -0.00693376 -0.0548285 -0.00283521  
 0.0264992 -0.0324526 0.0130489 0.0220633 0.0143932 0.0313198  
 -0.0332642 0.026298 -0.00218729 -0.0234872 0.01674 -0.0323937  
 -0.00120034 -0.00705026 0.0101072 0.000669908 0.0239051 0.000594439  
 0.0501077 -0.0102034 0.015353 0.0214981 0.00477263 0.0230773  
 -0.068429 0.0098568 -0.0280621 -0.014249 -0.00539748 -0.0495082  
 -0.00493663 0.00444968 -0.00373092 -0.0106548 -0.0112012 0.0259574  
 -0.0147425 0.00155167 -0.00824542 -0.000282709 -0.00239719 -0.0114156  
 -0.0213612 0.00496808 -0.0301667 0.0244387 -0.0238987 -0.0143243  
 -0.0348656 0.0137549 -0.020351 0.0273913 -0.032567 -0.0179187  
 0.038526 -0.0109208 0.0476918 -0.0492535 0.0403915 0.0385636  
 0.00325678 -0.00672411 -0.0253615 0.00772158 -0.00329346 -0.000882114  
 0.00799869 -0.0049376 -0.00265037 -0.00456757 -0.0410264 -0.0167602  
 -0.0312895 -0.0359448 -0.0297319 0.000716878 0.0217567 0.00524688  
 0.0141998 0.0469388 0.0232884 0.00630664 0.0172786 0.0251155  
 0.00726564 0.0162706 0.00690328 0.000899176 -0.00828548 0.00463934

(5) 纯二次拟合的方程为:

$$y_2 = 132.97 - 0.071475 x_1 + 0.14282 x_2 - 0.0032423 x_3 + 0.053676 x_4 + 0.066544 x_5$$

$$- 0.074163 x_6 - 0.034899 x_7 + 0.056922 x_8 + 5.9819e-005 x_1^2 - 0.00010169 x_2^2$$

$$+ 6.803e-006 x_3^2 - 0.00012452 x_4^2 + 7.0699e-005 x_5^2 - 0.00014541 x_6^2$$

$$+ 0.00011794 x_7^2 + 0.0002001 x_8^2;$$

$$\begin{aligned}
y_3 = & -122.65 - 0.056313 x_1 + 0.21712 x_2 - 0.12484 x_3 - 0.062224 x_4 + 0.15922 x_5 \\
& + 0.017201 x_6 + 0.029138 x_7 - 0.18845 x_8 - 4.3229e-005 x_1^2 - 0.00094704 x_2^2 \\
& - 7.6167e-005 x_3^2 + 0.00031895 x_4^2 - 0.00011937 x_5^2 - 4.7556e-005 x_6^2 \\
& - 0.00023629 x_7^2 - 5.4242e-005 x_8^2 ;
\end{aligned}$$

$$\begin{aligned}
y_4 = & 71.575 - 0.038648 x_1 - 0.087525 x_2 + 0.20414 x_3 - 0.011087 x_4 + 0.030305 x_5 \\
& - 0.0028316 x_6 + 0.17493 x_7 + 0.11456 x_8 + 1.5383e-005 x_1^2 - 9.2516e-005 x_2^2 \\
& + 2.5577e-006 x_3^2 - 5.8848e-005 x_4^2 - 0.00015202 x_5^2 + 3.2567e-005 x_6^2 \\
& - 0.00020713 x_7^2 - 0.00018543 x_8^2 ;
\end{aligned}$$

$$\begin{aligned}
y_5 = & 147.34 - 0.033249 x_1 + 0.14747 x_2 - 0.049513 x_3 - 0.036514 x_4 - 0.095619 x_5 \\
& - 0.0073129 x_6 - 0.044166 x_7 - 0.027962 x_8 + 0.00012105 x_1^2 + 0.00058468 x_2^2 \\
& - 3.8891e-005 x_3^2 - 2.9554e-005 x_4^2 + 0.00010634 x_5^2 + 0.000283 x_6^2 \\
& + 0.00028546 x_7^2 + 8.791e-005 x_8^2 ;
\end{aligned}$$

$$\begin{aligned}
y_6 = & 129.1 + 0.27123 x_1 - 0.17425 x_2 - 0.071478 x_3 + 0.11041 x_4 + 0.036497 x_5 \\
& - 0.052011 x_6 + 0.09531 x_7 - 0.0094375 x_8 - 0.00012532 x_1^2 + 0.00069995 x_2^2 \\
& - 1.7189e-005 x_3^2 - 0.00010676 x_4^2 + 3.471e-005 x_5^2 + 0.00018899 x_6^2 \\
& + 0.00049588 x_7^2 + 4.5136e-005 x_8^2 ;
\end{aligned}$$

附录三 关于确定潮流值和各机组出力关系的程序

(1) 用 Matlab 软件做多元二项式回归的程序:

```

A=[ 120 73 180 80 125 125 81.1 90
    133.02 73 180 80 125 125 81.1 90
    129.63 73 180 80 125 125 81.1 90
    158.77 73 180 80 125 125 81.1 90
    145.32 73 180 80 125 125 81.1 90
    120 78.596 180 80 125 125 81.1 90
    120 75.45180 80 125 125 81.1 90
    120 90.487 180 80 125 125 81.1 90
    120 83.848 180 80 125 125 81.1 90
    120 73 231.39 80 125 125 81.1 90

```

```

120 73 198.48 80 125 125 81.1 90
120 73 212.64 80 125 125 81.1 90
120 73 190.55 80 125 125 81.1 90
120 73 180 75.857 125 125 81.1 90
120 73 180 65.958 125 125 81.1 90
120 73 180 87.258 125 125 81.1 90
120 73 180 97.824 125 125 81.1 90
120 73 180 80 150.71 125 81.1 90
120 73 180 80 141.58 125 81.1 90
120 73 180 80 132.37 125 81.1 90
120 73 180 80 156.93 125 81.1 90
120 73 180 80 125 138.88 81.1 90
120 73 180 80 125 131.21 81.1 90
120 73 180 80 125 141.71 81.1 90
120 73 180 80 125 149.29 81.1 90
120 73 180 80 125 125 60.582 90
120 73 180 80 125 125 70.962 90
120 73 180 80 125 125 64.854 90
120 73 180 80 125 125 75.529 90
120 73 180 80 125 125 81.1 104.84
120 73 180 80 125 125 81.1 111.22
120 73 180 80 125 125 81.1 98.092
120 73 180 80 125 125 81.1 120.44];
x1=A(:,1);
x2=A(:,2);
x3=A(:,3);
x4=A(:,4);
x5=A(:,5);
x6=A(:,6);
x7=A(:,7);
x8=A(:,8);
B=[164.78 140.87 144.25 119.09 135.44 157.69
165.81 140.13 145.14 118.63 135.37 160.76
165.51 140.25 144.92 118.7135.33 159.98
167.93 138.71 146.91 117.72 135.41 166.81
166.79 139.45 145.92 118.13 135.41 163.64
164.94 141.5143.84 118.43 136.72 157.22
164.8141.13 144.07 118.82 136.02 157.5
165.59 143.03 143.16 117.24 139.66 156.59
165.21 142.28 143.49 117.96 137.98 156.96
167.43 140.82 152.26 129.58 132.04 153.6
165.71 140.82 147.08 122.85 134.21 156.23
166.45 140.82 149.33 125.75 133.28 155.09
165.23 140.85 145.82 121.16 134.75 156.77

```













```

132.04,134.21,133.28,134.75,135.57,135.97,135.06,134.67,133.75,134.27,
134.88,133.28,136.33,135.81,136.55,137.11,135.5,135.44,135.4,135.4,
135.28,135.16,135.29,135.12);
    double a6[34]={0,157.69,160.76,159.98,166.81,163.64,157.22,157.5,156.59,156.96,153.6,
156.23,155.09,156.77,157.2,156.31,158.26,159.28,158.83,158.37,158.01,
159.12,157.59,157.67,157.59,157.65,154.26,155.93,154.88,156.68,157.65,
157.63,157.61,157.64};
double y1[34],y2[34],y3[34],y4[34],y5[34],y6[34];
    double z1[34],z2[34],z3[34],z4[34],z5[34],z6[34];
    int i=0;
for(i=1;i<=33;i++)
    {
y1[i]=110.48+0.082607*x1[i]+0.047764*x2[i]+0.052794*x3[i]+0.11986*x4[i]
-0.025705*x5[i]+0.12165*x6[i]+0.12199*x7[i]-0.0015179*x8[i];
    y2[i]=131.35-0.054717*x1[i]+0.1275*x2[i]-0.00014644*x3[i]+0.033224*x4[i]
+0.086667*x5[i]-0.11269*x6[i]-0.018644*x7[i]+0.098528*x8[i];
    y3[i]=-108.99-0.069387*x1[i]+0.061985*x2[i]-0.1565*x3[i]-0.009871*x4[i]
+0.12467*x5[i]+0.0023561*x6[i]-0.0027873*x7[i]-0.20119*x8[i];
    y4[i]=77.612-0.034632*x1[i]-0.10278*x2[i]+0.20504*x3[i]-0.020882*x4[i]
-0.012018*x5[i]+0.0056932*x6[i]+0.14522*x7[i]+0.076336*x8[i];
    y5[i]=133.13+0.00032711*x1[i]+0.24283*x2[i]-0.06471*x3[i]-0.041202*x4[i]
-0.065452*x5[i]+0.070026*x6[i]-0.0038961*x7[i]-0.0091698*x8[i];
    y6[i]=120.85+0.23757*x1[i]-0.060693*x2[i]-0.078055*x3[i]+0.092897*x4[i]
+0.046634*x5[i]-0.00029128*x6[i]+0.16636*x7[i]+0.00038828*x8[i];
    +0.09531*x7[i]-0.0094375*x8[i]-0.00012532*pow(x1[i],2)+0.00069995*pow(x2[i],2)
-1.7189e-005*pow(x3[i],2)-0.00010676*pow(x4[i],2)+3.471e-005*pow(x5[i],2)+0.00018899*pow(x6[i],2)
+0.00049588*pow(x7[i],2)+4.5136e-005*pow(x8[i],2);
        cout<<y1[i]<<"\t";
        cout<<y2[i]<<"\t";
        cout<<y3[i]<<"\t";
        cout<<y4[i]<<"\t";
        cout<<y5[i]<<"\t";
        cout<<y6[i]<<"\t";
        cout<<"\n";
    }
}

```

(6) 搜索下一个时段各机组的出力分配预案的 C 语言程序

```

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<iostream>

#define INFINITY 100000
#define M 8

```

```
#define N 10

using namespace std;
int sp[M][N]={ -505,0,124,168,210,252,312,330,363,489,
               -560,0,182,203,245,300,320,360,410,495,
               -610,0,152,189,233,258,308,356,415,500,
               -500,150,170,200,255,302,325,380,435,800,
               -590,0,116,146,188,215,250,310,396,510,
               -607,0,159,173,205,252,305,380,405,520,
               -500,120,180,251,260,306,315,335,348,548,
               -800,153,183,233,253,283,303,318,400,800};
double sc[M][N]={ 70,0,50,0,0,30,0,0,0,40,
                 30,0,20,8,15,6,2,0,0,8,
                 110,0,40,0,30,0,20,40,0,40,
                 55,5,10,10,10,10,15,0,0,1,
                 75,5,15,0,15,15,0,10,10,10,
                 95,0,10,20,0,15,10,20,0,10,
                 50,15,5,15,10,10,5,10,3,2,
                 70,0,20,0,20,0,20,10,15,5};

void solve(int time,double load)
{
    int i,j,m=0,n=0,min=INFINITY;
    double remain=load,result=0;
    double scr[M][N]={0};
    double ramp[M]={2.2,1,3.2,1.3,1.8,2,1.4,1.8};
    double last[M]={120,73,180,80,125,125,81.1,90};
    double current[M]={0};
    while(remain>0)
    {
        min=INFINITY;
        for(i=0;i<M;i++)
            for(j=0;j<N;j++)
            {
                if(sp[i][j]<min)
                {
                    min=sp[i][j];
                    m=i;
                    n=j;
                }
            }
        if(current[m]+sc[m][n]>last[m]+time*ramp[m])
        {
```

```

scr[m][n]=last[m]+time*ramp[m]-current[m];
remain=remain-scr[m][n];
current[m]=current[m]+scr[m][n];
}
else
{
scr[m][n]=sc[m][n];
remain=remain-scr[m][n];
current[m]=current[m]+scr[m][n];
}
sp[m][n]=INFINITY+1;
}
scr[m][n]=scr[m][n]+remain;

for(i=0;i<M;i++)
current[i]=0;
for(i=0;i<M;i++)
for(j=0;j<N;j++)
current[i]=current[i]+scr[i][j];

cout<<"预测负荷 1052.8MW 时分配后各机组段容量"<<endl;
cout<<"机组/段 1 2 3 4 5 6 7 8 9 10 出力方案"<<endl;
for(i=0;i<M;i++)
{ cout<<i+1<<" ";
for(j=0;j<N;j++)
{ cout<<scr[i][j]<<" ";
}
cout<<current[i]<<" ";
cout<<endl;
}
cout<<endl;

for(i=0;i<M;i++)
printf("%.2f,",current[i]);

for(i=0;i<M;i++)
{
for(j=0;j<N;j++)
{result=result+scr[i][j];
}
}
cout<<result;
cout<<endl;

```

```
}
```

```
int main()
```

```
{
```

```
    int k;
```

```
    slove(15,1052.8);
```

```
    getch();
```

```
    return 0;
```

```
}
```

(7) 计算差值的 C 语言程序为:

```
#include "stdafx.h"
```

```
#include "iostream.h"
```

```
void main(int argc, char* argv[])
```

```
{
```

```
    double y1,x1,x2,x3,x4,x5,x6,y2[5],a11[5],a12[5],a13[5],a14[5],a15[5],a16[5];
```

```
    int i=0;
```

```
    a11[0]=0;
```

```
    a12[0]=0;
```

```
    a13[0]=0;
```

```
    a14[0]=0;
```

```
    a15[0]=0;
```

```
    a16[0]=0;
```

```
y2[0]=0;
```

```
    y1=120;
```

```
    x1=164.78;
```

```
    x2=140.87;
```

```
    x3=-144.25;
```

```
    x4=119.09;
```

```
    x5=135.44;
```

```
    x6=157.69;
```

```
    double y[5]={0,133.02,129.63,158.77,145.32};
```

```
    double a1[5]={0,165.81,165.51,167.93,166.79};
```

```
    double a2[5]={0,140.13,140.25,138.71,139.45};
```

```
    double a3[5]={0,-145.14,-144.92,-146.91,-145.92};
```

```
    double a4[5]={0,118.63,118.7,117.72,118.13};
```

```
    double a5[5]={0,135.37,135.33,135.41,135.41};
```

```
    double a6[5]={0,160.76,159.98,166.81,163.64};
```

```
for(i=1;i<=4;i++)
```

```
{
```

```
    y2[i]=y[i]-y1;
```

```
    a11[i]=a1[i]-x1;
```

```
    a12[i]=a2[i]-x2;
```

```
    a13[i]=a3[i]-x3;
```

```
        a14[i]=a4[i]-x4;
        a15[i]=a5[i]-x5;
        a16[i]=a6[i]-x6;
    }
    for(i=1;i<=4;i++)
    {
        cout<<y2[i]<<" \t ";
    }
    cout<<"\n";
    cout<<"-----\n";
    for(i=1;i<=4;i++)
    {
        cout<<a11[i]<<" \t ";
    }
    cout<<"\n";
    cout<<"-----\n";
    for(i=1;i<=4;i++)
    {
        cout<<a12[i]<<" \t ";
    }
    cout<<"\n";
    cout<<"-----\n";
    for(i=1;i<=4;i++)
    {
        cout<<a13[i]<<" \t ";
    }
    cout<<"\n";
    cout<<"-----\n";
    for(i=1;i<=4;i++)
    {
        cout<<a14[i]<<" \t ";
    }
    cout<<"\n";
    cout<<"-----\n";
    for(i=1;i<=4;i++)
    {
        cout<<a15[i]<<" \t ";
    }
    cout<<"\n";
    cout<<"-----\n";
    for(i=1;i<=4;i++)
    {
        cout<<a16[i]<<" \t ";
    }
}
```

```
cout<<"\n";
```

```
}
```

(8) 多项式拟合的 MATLAB 程序为:

```
>> y1=[1.03,0.73,3.15,2.01];
```

```
>> x1=[13.02,9.63,38.77,25.32];
```

```
>> a=polyfit(x1,y1,1)
```

```
a =
```

```
0.0826 -0.0608
```

```
>> x11=[9:1:39];
```

```
>> z1=a(2)+a(1)*x11;
```

```
>> y2=[-0.74,-0.62,-2.16,-1.42];
```

```
>> b1=polyfit(x1,y2,1)
```

```
b1 =
```

```
-0.0536 -0.0722
```

```
>> z2=z1+b1(2)+b1(1)*x11;
```

```
>> x12=[9:1:39];
```

```
>> z2=b1(2)+b1(1)*x12;
```

```
>> y3=[-0.89,-0.67,-2.66,-1.67];
```

```
>> b2=polyfit(x1,y3,1)
```

```
b2 =
```

```
-0.0680 0.0018
```

```
>> z3=b2(2)+b2(1)*x12;
```

```
>> y4=[-0.46,-0.39,-1.37,-0.96];
```

```
>> b3=polyfit(x1,y4,1)
```

```
b3 =
```

```
-0.0346 -0.0436
```

```
>> z4=b3(2)+b3(1)*x12;
```

```
>> subplot(3,2,1)
```

```
>> plot(x1,y1,'*')
```

```
>> hold on
```

```
>> plot(x11,z1,'-r')
```

```
>> subplot(3,2,2)
```

```
>> plot(x1,y2,'*')
>> hold on
>> plot(x12,z2,'-r')
>> subplot(3,2,3)
>> plot(x1,y3,'*')
>> hold on
>> plot(x12,z3,'-r')
>> subplot(3,2,4)
>> plot(x1,y4,'*')
>> hold on
>> plot(x12,z4,'-r')
>> y5=[-0.07,-0.11,-0.03,-0.03];
>> b5=polyfit(x1,y5,1)
```

b5 =

0.0025 -0.1141

```
>> z5=b5(2)+b5(1)*x12;
>> subplot(3,2,5)
>> plot(x1,y5,'*')
>> hold on
>> plot(x12,z5,'-r')
>> b52=polyfit(x1,y5,2)
```

b52 =

-0.0002 0.0113 -0.1954

```
>> z51=b52(3)+b52(2)*x12+b52(1)*x12.^2;
>> plot(x12,z51,'-k')
>> b53=polyfit(x1,y5,3)
```

b53 =

0.0000 -0.0012 0.0342 -0.3375

```
>> subplot(3,2,6)
>> y6=[3.07,2.29,9.12,5.95];
>> b6=polyfit(x1,y6,1)
```

b6 =

0.2345 0.0223

```
>> z6=b6(2)+b6(1)*x12;
```

```
>> plot(x1,y6,'*')
```

```
>> hold on
```

```
>> plot(x12,z6,'-r')
```

(9) 用 Lingo 软件对初始出力方案进行优化，程序代码和结果如下：

model:

```
max=(320*(x2-79)+308*(x3-180)+305*(x6-140)-(303-252)*(150-x1)-(303-302)*(99.5-x4)-(303-215)*(125-x5)-(303-260)*(95-x7)-(303-303)*(113.9-x8))/4;
```

```
120.85+0.23757*x1-0.060693*x2-0.078055*x3+0.092897*x4+0.046634*x5-0.00029128*x6+0.16636*x7+0.00038828*x8<162;
```

```
120<x1;x1<153;
```

```
58<x2;x2<81;
```

```
132<x3;x3<180;
```

```
90<x4;x4<99.5;
```

```
110<x5;x5<142;
```

```
125<x6;x6<150;
```

```
85<x7;x7<102.1;
```

```
110<x8;x8<117;
```

```
x1+x2+x3+x4+x5+x6+x7+x8=982.4;
```

```
end
```

Global optimal solution found at iteration: 10

Objective value: 932.2702

Variable	Value	Reduced Cost
X2	81.00000	0.000000
X3	180.0000	0.000000
X6	150.0000	0.000000
X1	120.0000	0.000000
X4	95.60998	0.000000
X5	142.0000	0.000000
X7	96.79002	0.000000
X8	117.0000	0.000000

Row	Slack or Surplus	Dual Price
1	932.2702	1.000000

---

2	0.000000	142.9291
3	0.000000	8.177981
4	33.00000	0.000000
5	23.00000	0.000000
6	0.000000	101.7025
7	48.00000	0.000000
8	0.000000	101.1840
9	5.609977	0.000000
10	3.890023	0.000000
11	32.00000	0.000000
12	0.000000	28.36233
13	25.00000	0.000000
14	0.000000	89.31932
15	11.79002	0.000000
16	5.309977	0.000000
17	7.000000	0.000000
18	0.000000	12.97219
19	0.000000	-13.02768

### 参考文献

- [1] 赵静、但琦主编，数学建模与数学实验，北京：高等教育出版社出版社，2000
- [2] 李林川，茅波等，基于统一边际电价结算的电力市场下考虑消除阻塞费用最小的竞价算法，电网技术，第28卷第7期，2004.4
- [3] 王秀丽，甘志等，输电阻塞管理的灵敏度分析模型算法，电力系统自动化，2002