

基于线性模型的电力系统储能调度优化问题研究

摘 要：

按照国家“双碳”目标，中国计划与2035年基本建成新型电力化系统，目前新型储能系统仍在商业化初期阶段。目前工商业储能公司主要通过分时电价带来的峰谷套利来进行获利，现要求通过历史数据预测未来的负荷数据来帮助公司获取更好的收益。

针对问题一，首先将时间、储能用电和市电三类数据处理为以15秒为间隔的数据，通过观察发现数据集中存在**缺失数据**，对于少部分数据采用**周期样条插值**进行数据补全，大部分缺失数据使用**Prophet预测模型**将其补全。并且将**时间频率间隔不为15s**的样本数据插入最近的时间频率间隔为15s且属性缺失的数据中。最后基于**历史负荷数据**采用**Prophet模型**、**LSTM模型**和**ARIMA模型**以15分钟为周期预测了三个储能设备未来一天的负荷数据，经评估比较后单柜、二并柜和三并柜储能设备所选择的模型分别为**Prophet模型**、**Prophet模型**、**ARIMA模型**，各模型预测结果的 R^2 值分别为**0.8498**、**0.9296**、**0.6451**。最终预测结果保存至对应的预测结果.xlsx的对应位置中。

针对问题二，首先根据题目要求建立在尽可能**降低需量电价成本**的前提下**最大化峰谷套利收益**的**双目标线性规划模型**，其次设计了基于**Gurobi求解器**的求解程序分别根据预测和真实数据进行求解，得到**帕累托前沿解集**，最后分别根据德尔菲法确定各储能系统的每分钟储能系统的**充放电方案**。所确定的各储能策略的**最优峰谷套利收益**为**332.07**、**785.81**、**1186.84元**，**需量电价成本**为**25672.96**，**33317.87**，**35263.47元**，**合同最大需量**为**534.85KW**、**694.12KW**、**734.66KW**。与原储能方案相比较，当天**峰谷套利收益**分别增加了**54.97元**、**159.2元**、**244.4元**，每月**需量电价成本**减少了**868.16元**、**1665.89元**、**1763.17元**。三个储能设备的具体方案放入附件预测结果.xlsx的对应位置中。

针对问题三，在问题二的基础上修改目标函数以及最大需量电价，从而得到新的**多目标线性规划模型**，其次设计了基于**Gurobi求解器**的求解程序根据预测数据进行求解，得到**帕累托前沿解集**，最后分别根据德尔菲法确定各储能系统的每分钟储能系统的**充放电方案**。所确定的各系统的**最优峰谷套利收益**为**332.07**、**785.81**、**1186.84元**，**需量电价成本**为**25672.96**，**33317.87**，**35263.47元**，**合同最大需量**为**534.85KW**、**694.12KW**、**734.66KW**，且各系统的最大需量均**不超过**合同最大需量的105%。与原储能策略相比较，各储能设备的当天**峰谷套利收益**分别增加了**54.97元**、**159.2元**、**244.4元**，每月**需量电价成本**减少了**868.16元**、**1665.89元**、**1763.17元**。各储能策略的具体方案放入附件预测结果.xlsx的对应位置中。

关键词：Prophet模型，ARIMA模型，线性规划，多目标，Gurobi

目录

一、问题重述	1
1.1问题背景	1
1.2问题提出	1
二、模型假设及符号说明	3
2.1模型假设	3
2.2符号说明	3
三、问题一模型建立与求解	4
3.1问题分析	4
3.2数据预处理	4
3.2.1数据分析	4
3.2.2异常数据处理	5
3.2.3数据补全	5
3.2.4对数变换	5
3.3 数据预测	5
3.3.1 单柜数据预测	5
3.3.2 二并柜数据预测	6
3.3.3 三并柜数据	7
四、 问题二模型建立与求解	9
4.1 问题分析	9
4.2 模型建立	9
4.2.1 决策变量	9
4.2.2 目标函数	9
4.2.3 约束条件	9
4.3 问题求解	11
4.3.1 基于加权和法与Gurobi的多目标模型求解	11
4.3.2单柜方案求解	12
4.3.3二并柜方案	13
4.3.4三并柜方案	15
五、 问题三模型建立与求解	17
5.1问题分析	17
5.2模型建立	17
5.2.1 目标函数	17
5.2.2 最大需量约束	17
5.2.3 整体模型	17
5.3 问题求解	18
六、模型评价	19
6.1 模型的优点	19
6.2 模型的缺点	19

6.3 模型的推广与改进	19
参考文献	20
附录	21

一、问题重述

1.1 问题背景

在国家“双碳”目标和电力发展规划的指导下，中国计划在2035年基本建成新型电力系统，并在2050年全面建成。目前，新型储能系统正处于商业化初期向规模化发展的阶段。当前，工商业储能公司主要通过利用分时电价的峰谷差异获利，即在低电价时段存储电能，在高电价时段释放电能出售。我国工商业用户的电价由多项费用构成，其中包括市场化购电电费、输配电量电费以及基本电费等。需量电费是根据用户最大需量来计算的，通常每月按最大需量进行结算，超过合同确定值105%的部分需按更高的电费标准收取。

储能公司在进行峰谷套利之前，需通过对历史负荷数据的预测，制定有效的储能策略，以实现更大的经济效益。负荷是指储能公司在每个时刻必须使用的电量，包括为其他用户供电及自用的需求功率，准确预测负荷数据是制定储能策略的关键。

1.2 问题提出

在本题中，储能公司的基本电费主要考虑合同最大需量方式计算的需量电费。需量是一个指定时间间隔内的平均功率，指定的时间间隔就是需量周期，我们国家通常采用 15 分钟做为需量周期，采用“滑差”的方式测量需量^[1]。其简易计算方法如图所示：

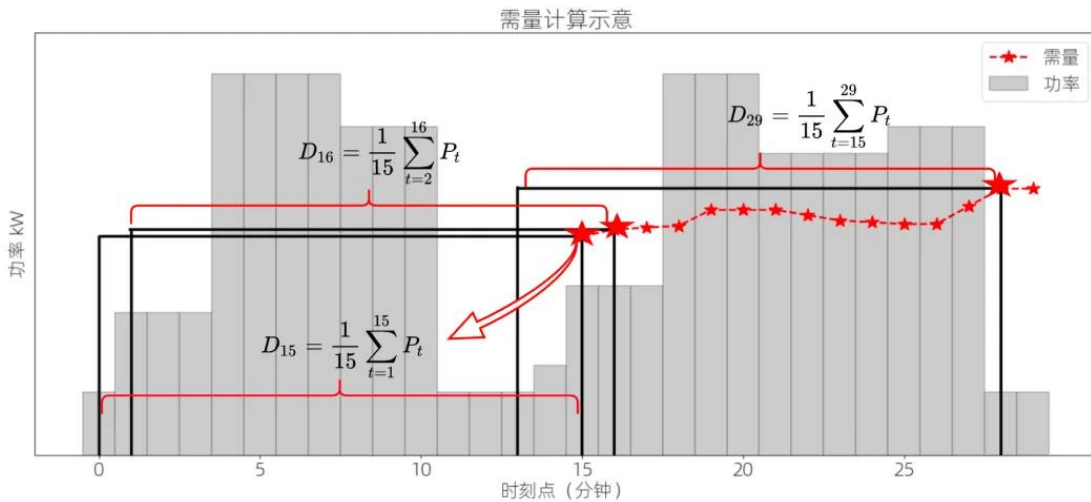


图 1-1 需量计算方式

图中P 表示功率，D 表示需量。通常一个月为一个结算周期，取每个月内的最大需量进行需量电费的计算。使用合同最大需量方式计算时，当最大需量超过合同确定值的105%时，超过部分按照基本电费价一倍计算；未超过合同确定值105%时，按合同确定值收取^[2]。

基于上述研究背景，本题目提供了5个附件，包括浙江省近几年的各个时段的分时电价数据和历史用电数据。为了优化储能系统的收益，本文需要根据上述附件解决以下三个问题：

问题一：负荷的预测对于制定储能系统的充放电策略至关重要，通过准确的负荷

预测可以提高储能系统的经济效益。现基于历史电力数据，以15分钟为周期、15秒为间隔，对未来一天三个储能设备的负荷进行预测。

问题二：在满足实际负荷需求的前提下，基于预测的负荷数据，建立并优化三个储能系统的充放电策略，目标是最大化峰谷套利收益并降低需量电价成本。将优化后的策略结果与原有充放电策略的收益进行比较，计算出新的策略所带来的收益提升。

问题三：假设最大需量可以超过设定的合同最大需量的105%，重新建模并设计三个储能设备的充放电策略，以进一步提高收益。在这一假设条件下，探索更为灵活的策略，以最大化峰谷套利收益并降低需量电价成本。最终，将新策略的结果与原有策略进行比较，分析收益的提升幅度。

二、模型假设及符号说明

2.1 模型假设

- 1、假设题目所给出的原始数据是真实有效的；
- 2、假设储能系统输电传输损耗为 0。

2.2 符号说明

表2-1 符号说明

符号	说明
x_i	时间点 i 储能装备充放电功率
s_i	时间点 i 的市电功率
H	合同最大需量
N	最大需量
R	峰谷套利收益
D	需量电价
V	储能系统的容量上限
h_i	时间点 i 的预测负荷功率
U	储能系统功率上限
P_i	P_i 表示时间点 i 的电价

三、问题一模型建立与求解

3.1 问题分析

问题一要求根据附件中储能公司电柜的历史用电记录来三个储能设备未来一天的负荷。通过观察数据发现存在部分缺失值以及部分数据的时间频率存在异常，因此需要对历史数据进行预处理，然后基于处理后数据建立预测模型，从而预测三个储能设备未来一天的负荷数据。

3.2 数据预处理

3.2.1 数据分析

本次分析所使用的数据为2023年11月3日至2024年3月17日的历史用电数据，分为单柜、二并柜、三并柜三个部分,其中主要包含时间数据(“Time”)、储能用电数据(“CN”)和市电数据(“SD”)。

为更好的捕捉负荷数据中的平滑趋势和季节性波动，将Time、CN和SD三类数据处理为以15秒为间隔的数据样本，经可视化处理后三部分的的数据分别如下图：

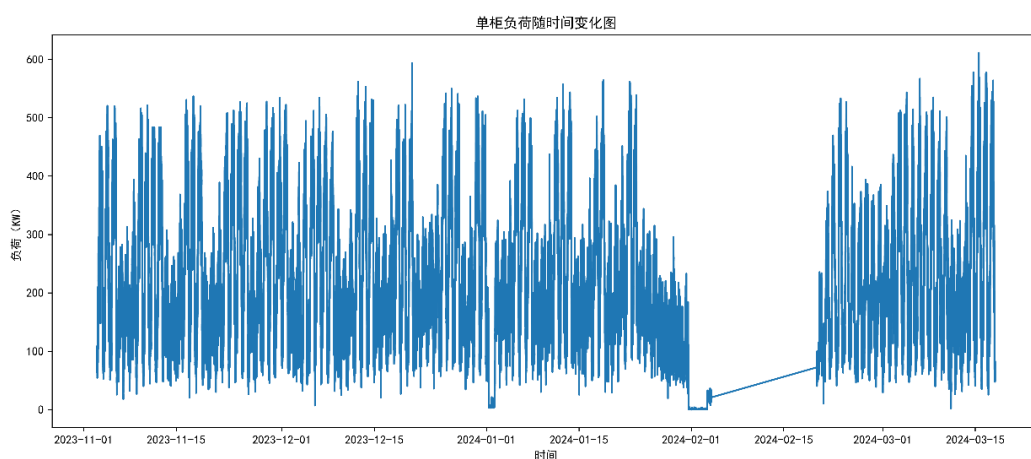


图1 单柜负荷数据随时间变化图

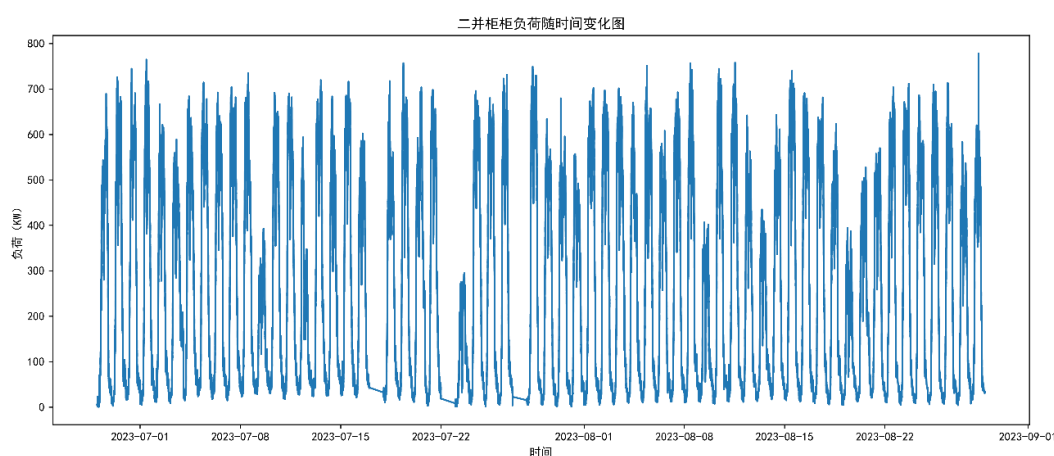


图2 二并柜负荷数据随时间变化图

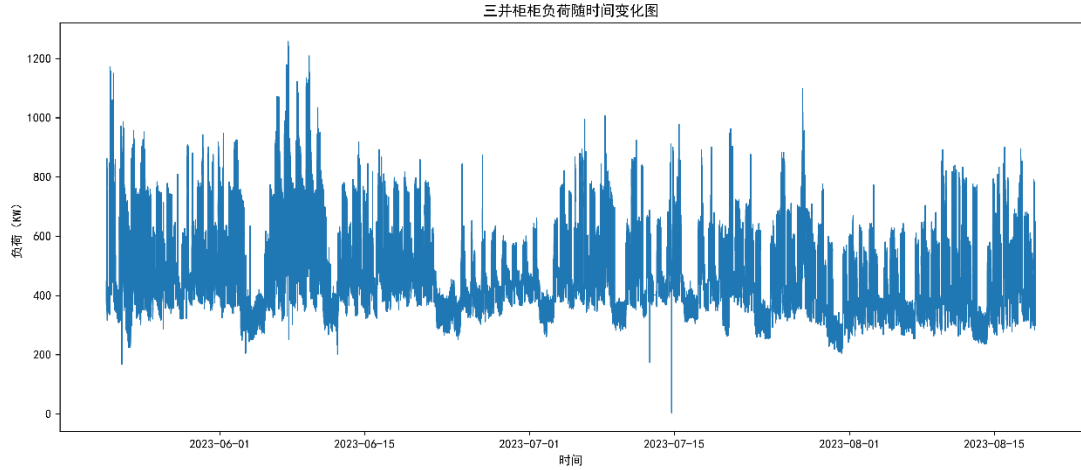


图3 三并柜负荷数据随时间变化图

观察图1、图2和图3以及具体负荷数据发现单柜数据、二并柜数据和三并柜数据都有一定的缺失。其中单柜数据的缺失值最多。单柜数据集中2024年2月4号到2024年2月19号的数据全部缺失，其他数据零散缺失。除数据缺失外还发现三个数据集中存在时间戳不为15秒间隔的异常数据。

3.2.2 异常数据处理

我们将时间戳不服从15秒间隔的异常数据进行处理。如果其异常数据时间的服从15秒间隔的上限或下限时间所对应的数据样本是缺失的，那我们将其时间数据填入缺失数据部分。

3.2.3 数据补全

为将缺失数据进行补全，小部分的缺失数据我们使用周期样条插值法将其补全。对于大部分的缺失数据例如单柜数据集中的2024年2月4号到2024年2月19号的数据完全缺失，我们使用Prophet预测模型补全其缺失数据。

3.2.4 对数变换

为防止预测的负荷数据为负，将历史数据进行对数变换，这样能够保证预测值在反变换后均为正数，其变换公式如下：

$$y = \log(y + c) \quad (3-1)$$

其中 y 表示原始数据， c 表示一个较小常数，以避免对数为负值。本题中的 c 为1。

3.3 数据预测

对于三个储能系统数据集的预测，分别使用Prophet^[4]、ARIMA^[7]和LSTM预测模型进行预测效果对比，最终选用效果最好的模型进行未来一天负荷数据的预测。

3.3.1 单柜数据预测

预测单柜数据时由于补全数据的范围太大所以导致补全部分的数据误差较大，所以我们选用了全部补全了后的单柜数据与没有补全的2024年2月20日后的单柜数据进行预测效果对比，最终由于2024年2月20日后数据的预测效果更优，因此我们最终选用2024年2月20日后的数据进行预测。

由于LSTM模型预测结果波动太小不符合实际情况，故不考虑其预测结果。通过观察图3-1发现Prophet模型的单并柜预测数据比较符合真实数据情况。因此最终选择Prophet模型对单柜未来一天的负荷数据进行预测。

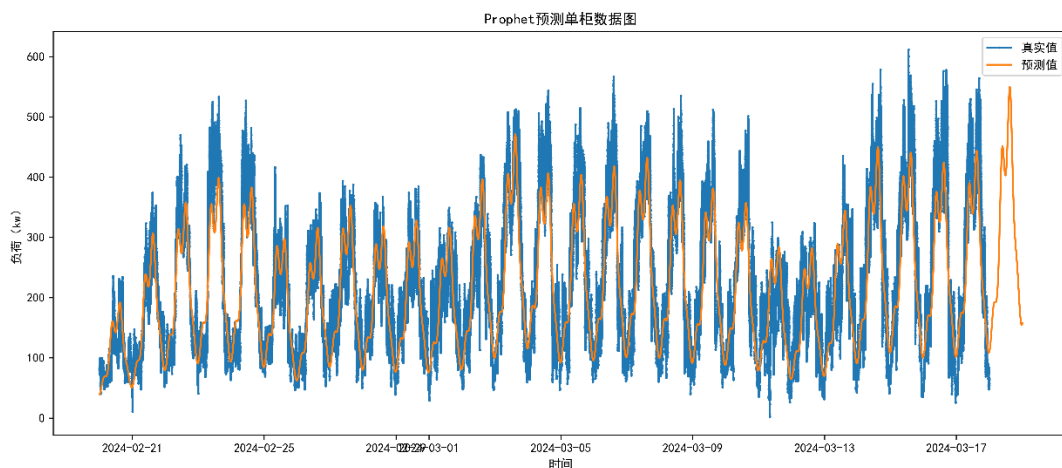


图 3- 1 Prophet 模型的单柜预测数据图

Prophet预测模型的评估结果如表3-1，结果表明单柜储能设备的Prophet预测模型的预测效果较优。

表3- 1单柜的Prophet预测模型评估结果

评估指标	Prophet模型指标值
MSE	2911.8643
MAE	43.7235
MAPE	22.5921%
R^2	0.8498

最终得到单并柜2023年8月29日的预测结果见图3-2可以发现模型拟合曲线符合真实数据情况。具体预测数据见附件“预测结果_单柜_2024-3-18.Xlsx”。

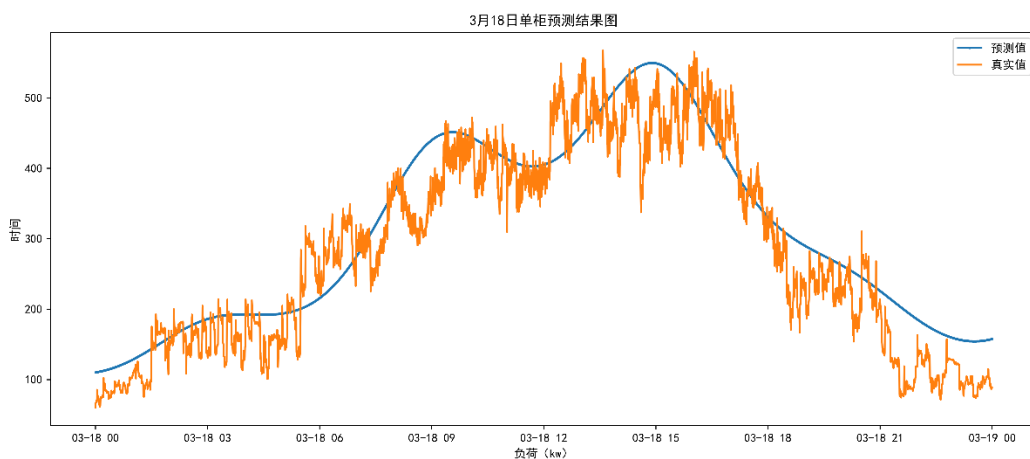


图 3- 2 Prophet 模型的未来一天单柜预测数据

3. 3. 2 二并柜数据预测

二并柜的Prophet模型数据预测结果见图3-3，通过观察发现其预测的数据波动和趋势符合原数据，说明模型预测的情况较优。LSTM的预测结果波动太小不符合数据波动情况，因此选择Prophet模型对二并柜未来一天的负荷数据进行预测。

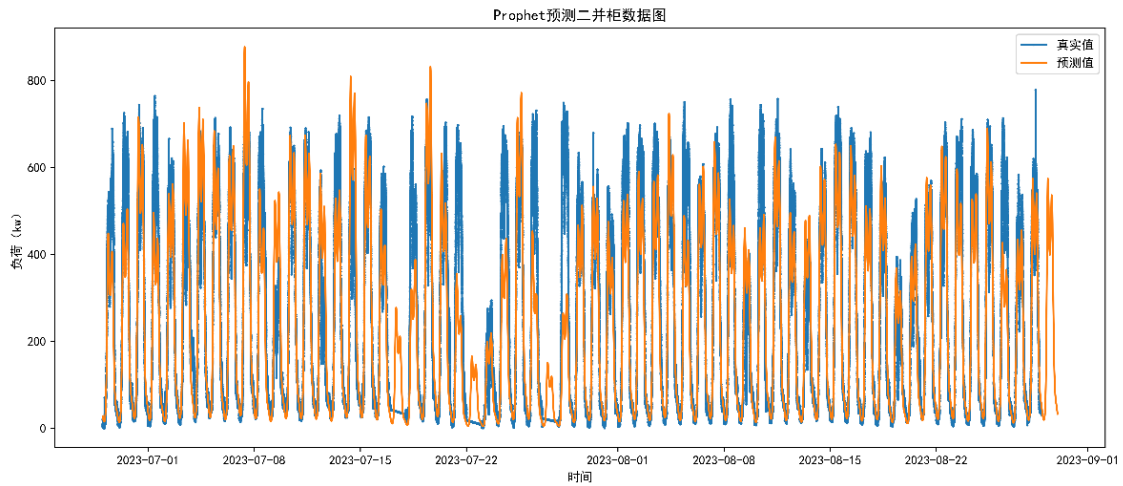


图3- 3 Prophet模型的二并柜预测数据图

Prophet预测模型的评估结果如表3-2, 结果表明二并柜储能设备的Prophet预测模型评估效果较好。

表3- 2二并柜的Prophet预测模型评估结果

评估指标	指标值
MSE	16682.7255
MAE	82.6515
MAPE	20.0499%
R^2	0.9296

二并柜2023年8月29日的预测结果见下图：

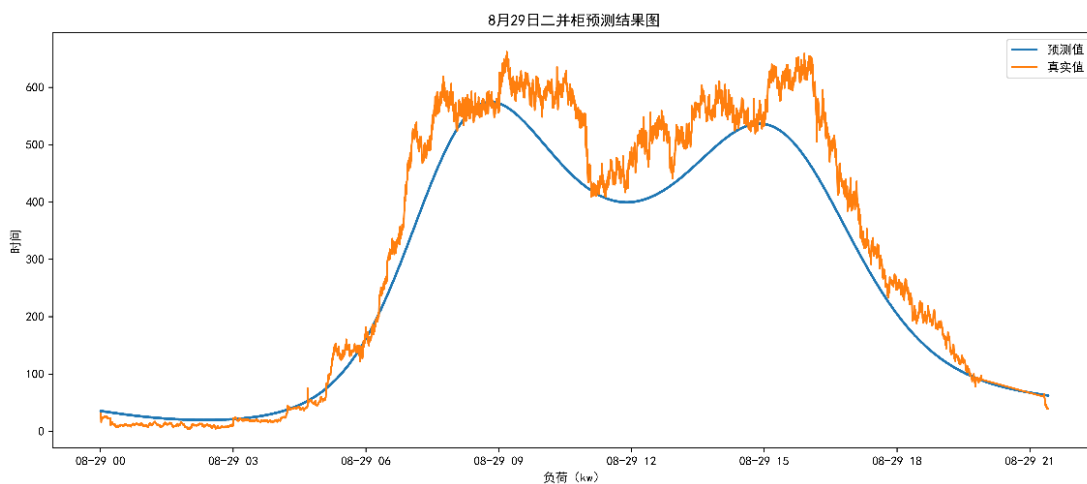


图3- 4 Prophet模型的未来一天二并柜预测数据图

观察图3-4发现8月29日负荷数据拟合曲线符合真实数据情况。具体预测数据见附件“预测结果_二并柜_2023-8-29.Xlsx”。

3. 3. 3 三并柜数据

三并柜的ARIMA模型数据预测结果见图3-5, 通过观察发现其预测的数据波动和趋势符合原数据, 说明模型预测的情况较优。除了ARIMA模型外我们还使用了LSTM和Prophet模型进行预测但是最终效果的评估结果较差, Prophet和LSTM无法拟合出波

动如此大的数据，因此我们选择ARIMA模型对三并柜未来一天的负荷数据进行预测。

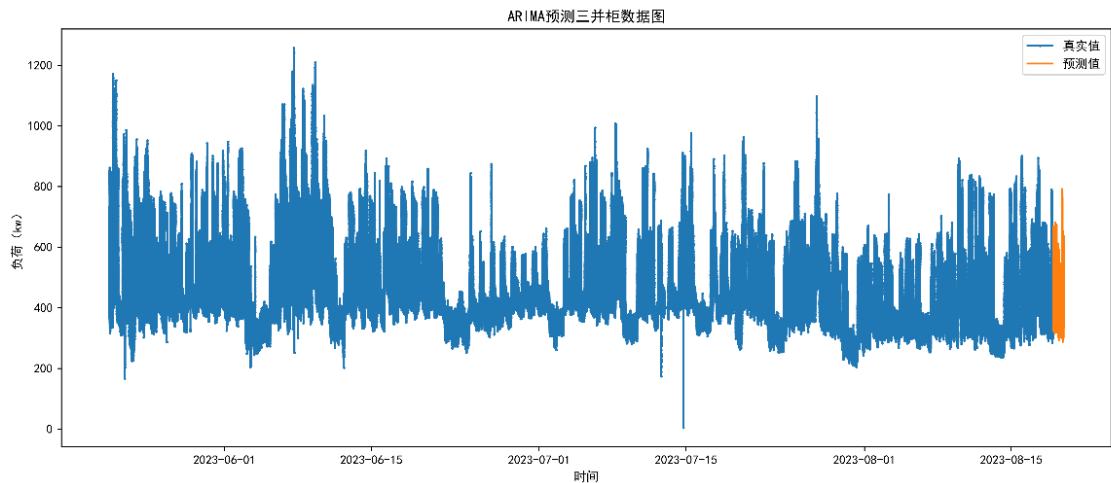


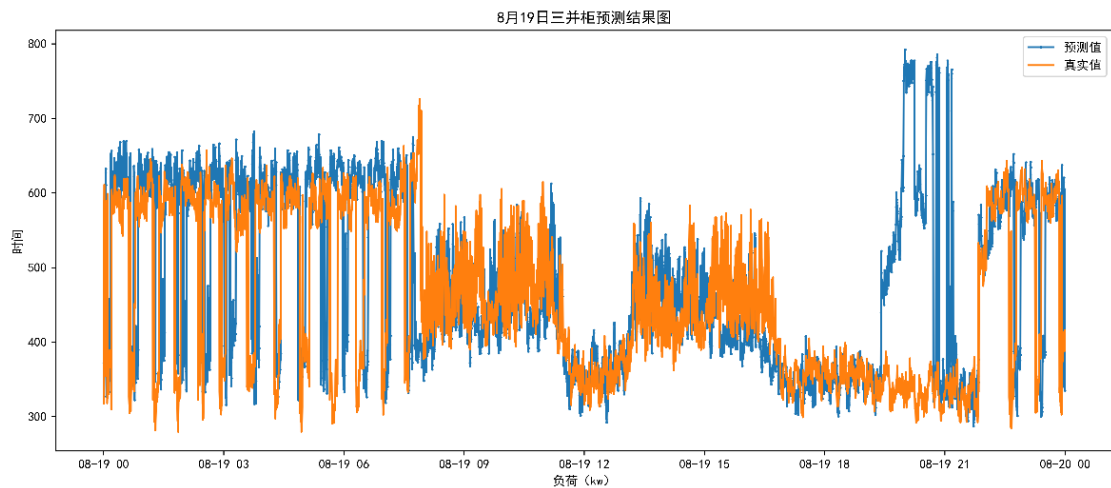
图3- 5 ARIMA模型的三并柜预测数据图

ARIMA预测模型的评估结果如表2，结果表明模型效果较好。

表3- 3三并柜的ARIMA预测模型评估结果

评估指标	指标值
MSE	16385.5627
MAE	102.8153
MAPE	20.2576%
R^2	0.6451

三并柜2023年8月29日的预测结果见下图：



观察图3-6发现8月29日负荷数据拟合曲线符合真实数据情况。具体预测数据见附件“预测结果_三并柜_2023-8-29.xlsx”。

四、问题二模型建立与求解

4.1 问题分析

问题二要求并根据预测后数据优化储能系统的充放电策略从而最大化总收益，并与未调整前的充放电策略进行比较。首先假定储能功率、市电功率在每分钟内不变，实际负荷功率为该分钟内每15秒负荷功率的平均值。然后将一天划分为1440分钟，建立双目标线性模型，求出每分钟储能系统的充放电策略和合同最大需量，并与原方案的收益进行比较。

4.2 模型建立

4.2.1 决策变量

(1) 决策变量 x_i

表示时间点 i 储能装备充放电功率，其中正值表示充电功率，负值表示放电功率：

$$x_i \in R, i \in T \quad (4-1)$$

其中 T 表示时间集合， $T = \{1, 2, 3, \dots, 1440\}$ ， T 中元素代号为 i, j 。

(2) 决策变量 s_i

表示第 i 分钟的市电功率：

$$s_i \geq 0, i \in T \quad (4-2)$$

(3) 决策变量 H

表示合同最大需量：

$$H \geq 0 \quad (4-3)$$

(4) 辅助变量 N

表示最大需量：

$$N \geq 0 \quad (4-4)$$

(5) 辅助变量 R

表示峰谷套利收益：

$$R \geq 0 \quad (4-5)$$

4.2.2 目标函数

根据题目要求，所建立的目标函数为在尽可能降低需量电价成本的前提下最大化峰谷套利收益：

$$\begin{aligned} \text{Max obj1} &= R \\ \text{Max obj2} &= -H \times D \end{aligned} \quad (4-6)$$

其中 D 表示需量电价。

4.2.3 约束条件

通过分析问题，我们所建立的约束条件如下：

1. 储能系统容量约束：

(1) 储能系统内电量不为负值：

$$\sum_{i=1}^j x_i \geq 0; j \in T \quad (4-7)$$

(2) 储能系统内电量不超过容量限制:

$$\frac{1}{60} \sum_{i=1}^j x_i \leq V; \quad j \in T \quad (4-8)$$

其中 V 表示该储能系统的容量上限, 因为 x_i 表示第 i 分钟平均功率, 所以第 i 分钟电量变化应当为 $\frac{1}{60}x_i \text{kw} \cdot \text{h}$; 式(4-3)由于右侧为 ≥ 0 , 所以约掉了 $\frac{1}{60}$ 。

2. 负荷、市电和储能用电三者关系约束

由题目所给条件 $\text{负荷} = \text{市电} - \text{储能用电}$, 可得以下约束。

$$s_i - x_i = h_i; \quad i \in T \quad (4-9)$$

其中 s_i 表示时间点 i 的市电功率, h_i 表示第 i 分钟的预测负荷功率。

3. 最大需量约束

(1) 由滑差法根据市电求出每个周期需量小于等于实际最大需量:

$$\frac{1}{15} \sum_{i=j}^{j+14} s_i \leq N, \quad j \in [1, 1426] \quad (4-10)$$

注: 第1427-1440分钟根据滑差法计算最大需量需要用到后一天前十五分钟数据, 但因为我们没有这些数据, 所以没有进行计算

(2) 最大需量不超过合同最大需量的105%:

$$N \leq 1.05 \times H \quad (4-11)$$

(3) 最大需量不低于当月已知最大需量:

由于需量电费是由当月实际最大需量与合同最大需量共同决定, 因此实际最大需量需要还需要大于月初到目标日(需要求出方案的这一天)前一天的最大需量 Np , Np 可以根据题目所给数据计算得到。目标日之后天数的最大需量对需量电费的影响我们不考虑。

$$N \geq Np \quad (4-12)$$

4. 充放电功率约束:

充放电功率不能超过储能系统上限

$$-U \leq x_i \leq U \quad (4-13)$$

其中 U 表示储能系统功率上限。

5. 峰谷套利收益:

峰谷套利收益可由下式计算得到, P_i 表示时间点 i 的电价。

$$R = -\frac{1}{60} \sum_{i \in T} P_i \times x_i \quad (4-14)$$

得到最终模型如下:

$$\begin{aligned}
& \text{Max obj1} = R \\
& \text{Max obj2} = -H \times D \\
& s. t. \begin{cases} \sum_{i=1}^j x_i \geq 0; j \in T \\ \frac{1}{60} \sum_{i=1}^j x_i \leq V; j \in T \\ s_i - x_i = h_i; i \in T \\ \frac{1}{15} \sum_{i=j}^{j+14} s_i \leq N; j \in [1, 1425] \\ N \leq 1.05 \times H \\ N \geq Np \\ -U \leq x_i \leq U; i \in T \\ R = -\frac{1}{60} \sum_{i \in T} P_i \times x_i \\ x_i \in R, i \in T \\ s_i \geq 0, i \in T \\ H \geq 0 \\ N \geq 0 \\ R \geq 0 \end{cases}
\end{aligned}$$

4.3 问题求解

4.3.1 基于加权和法与Gurobi的多目标模型求解

1. 求解思路

上文建立的模型是一个多目标模型，对于多目标问题，可通过加权和法转化为多个单目标问题，即对目标1和目标2加上一组权重值进行求解。又因为将多目标转为单目标后的模型是一个所有变量都为连续变量的线性规划，这是一个**NP完成**问题，可以通过求解器迅速求出最优解。因此我们可以设计python程序调用用Gurobi求解器对以上模型进行求解，得到一个由于目标权重不同导致最优结果差异的帕累托前沿解集，并通过德尔菲法从中选择最优方案。

2. 求解过程

(1) 首先对已知数据进行处理，可以得到各储能系统的具体参数值见表4-1，其余参数值如各时段电价与实际负荷数据限于篇幅详见附件。

表4-1 各储能系统参数值

参数	单柜数值	二并柜数值	三并柜数值
需量电价	48	48	48
当月已知最大需量	552.9400	728.8283	771.3883

(2) 其次通过试验确定加权和法的权重方案为：目标一（峰谷套利收益）权重值恒定为1，目标二（需量电价成本）权重值从0.001开始，以0.001为步长逐步增至0.030，共计30组权重值，分别对模型进行求解，得到帕累托前沿解集。

(3) 最后使用德尔菲法从帕累托前沿解集中选择最终方案。

4.3.2 单柜方案求解

1、基于Gurobi求帕累托前沿解集

将单柜储能系统的具体数据带入多目标线性模型，为了对预测结果进行验证分别使用了真实数据和上一题预测的数据进行求解，基于Gurobi求解器求得的帕累托前沿解集见图4-1和图4-2：

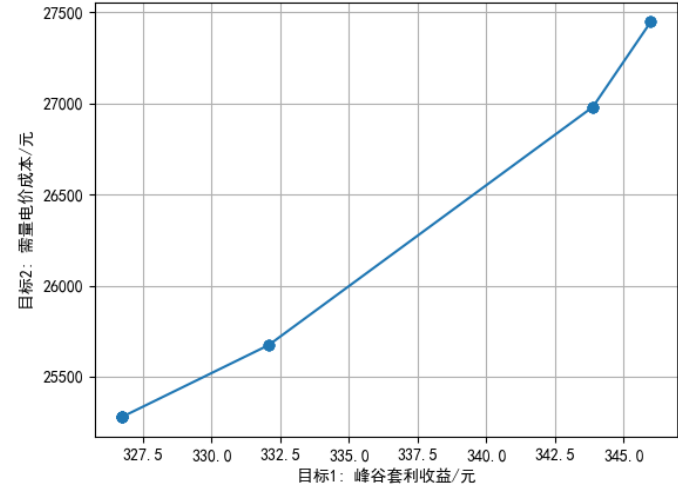


图 4-1 真实数据单柜系统帕累托前沿解集图

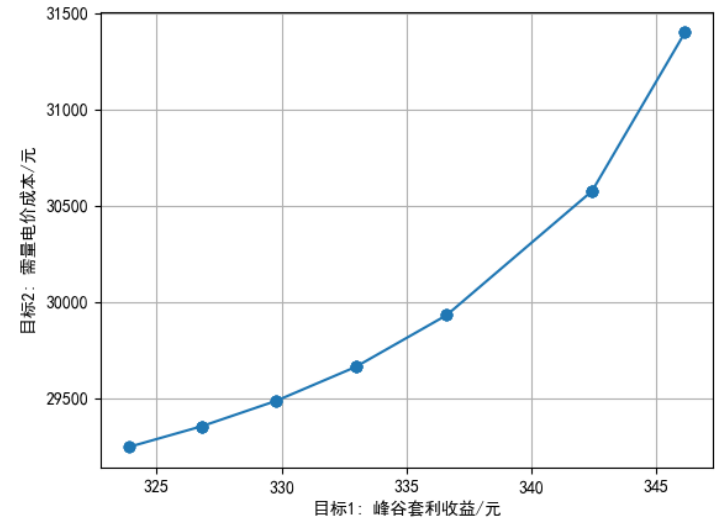


图4-2 预测数据单柜系统帕累托前沿解集图

观察基于真实数据和预测数据的帕累托前沿解集图，发现两者的峰谷套利收益区间差距不大，但是预测数据的需量电价成本较真实结果有一个较大升高，由此可以看出我们很好地预测出了负荷的变化趋势，但是预测的数值会略大于真实值，这也真实值与预测值的对比图的内容一致。

2、基于德尔菲法选择最终方案

采用德尔菲法选择的单柜储能系统的优化策略，真实数据与预测数据的策略收益见表4-2，单柜的优化策略见图4-3，具体储能方案见附件。

表4- 2单柜储能策略表

数据	峰谷套利收益/元	需量电价/元	合同最大需量/KW
真实数据	332.07	25672.96	534.85
预测数据	342.42	30574.92	636.98

与原策略相比，当日的峰谷套利收益增加54.97元，月需量电费减少868.16元。

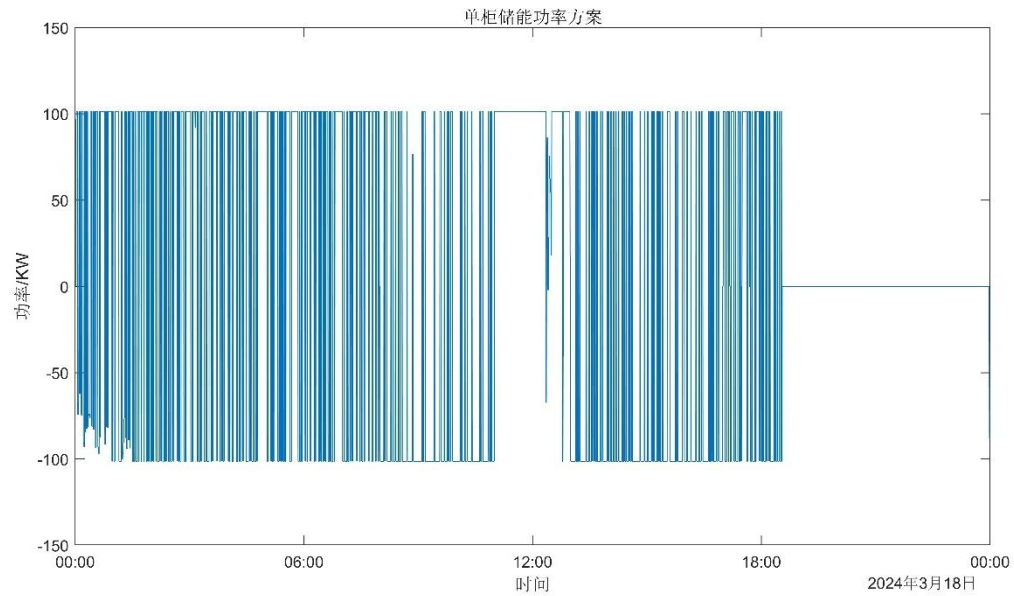


图4- 3单柜储能设备优化策略图

4. 3. 3 二并柜方案

1、基于Gurobi求帕累托前沿解集

将单柜储能系统的真实数据和预测数据数据代入多目标线性模型，基于Gurobi求解器求得的帕累托前沿解集见图4-3和图4-4：

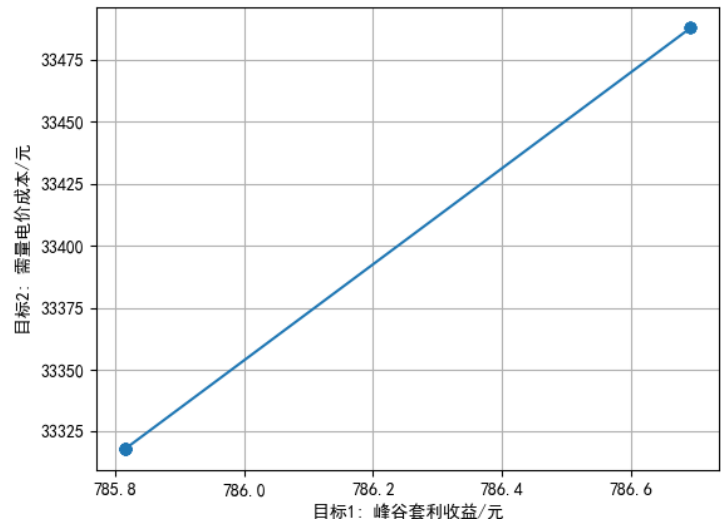


图4-4真实数据二并柜系统帕累托前沿解集图

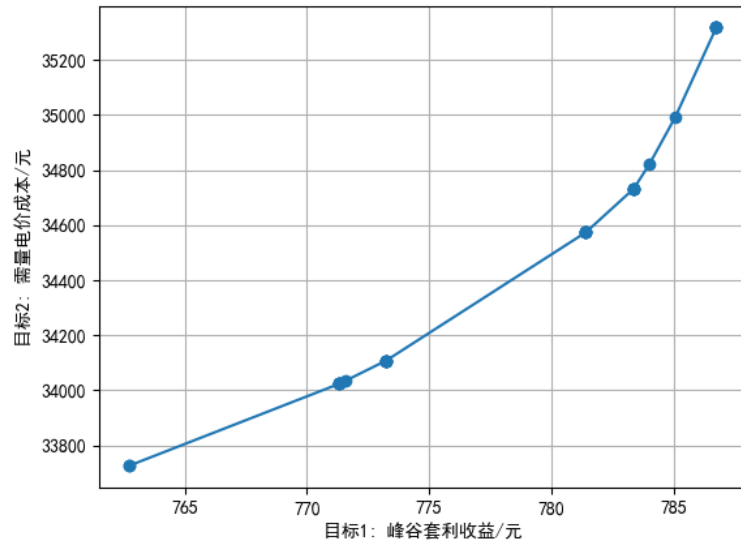


图4-5预测数据二并柜系统帕累托前沿解集图

对比图4-3和图4-4, 和单柜数据一样, 可以看出预测数据很好地预测了负荷趋势, 但较实际数值偏大一些。

2、基于德尔菲法选择最终方案

采用德尔菲法选择的单柜储能系统的优化策略, 真实数据与预测数据的策略收益见表4-3, 单柜的优化策略见图4-5, 具体储能方案见附件。

表4-3二并柜储能策略收益表

数据	峰谷套利收益/元	需量电价/元	合同最大需量/KW
真实数据	785.81	33317.87	694.12
预测数据	785.07	34991.63	728.99

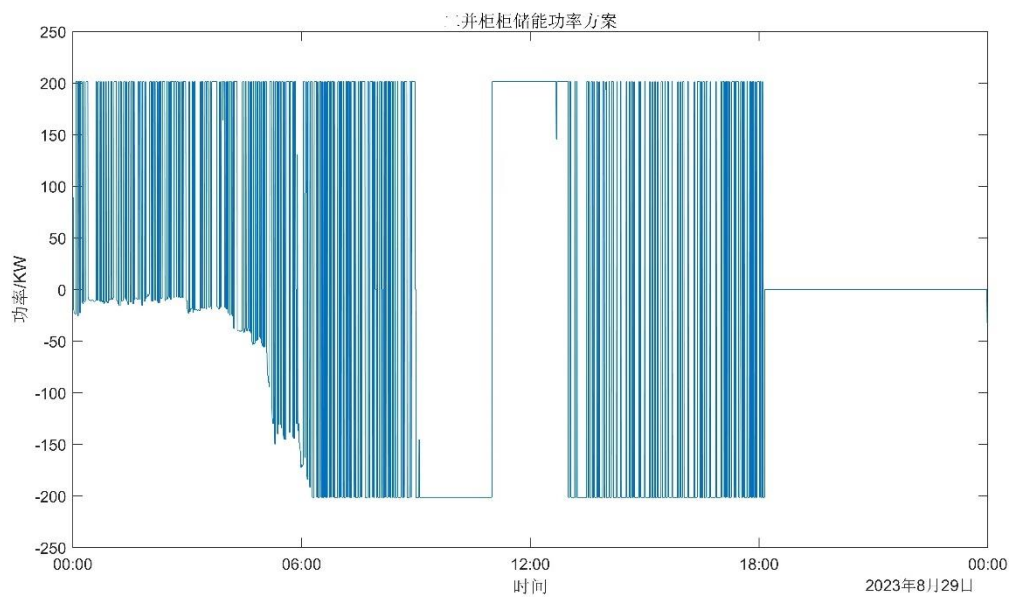


图4-6二并柜储能设备优化策略图

与原策略相比，当日的峰谷套利收益增加159.2元，月需量电费减少1665.89元。

4.3.4 三并柜方案

1、

将单柜储能系统的具体数据带入多目标线性模型，基于Gurobi求解器求得的帕累托前沿解集见图4-7：

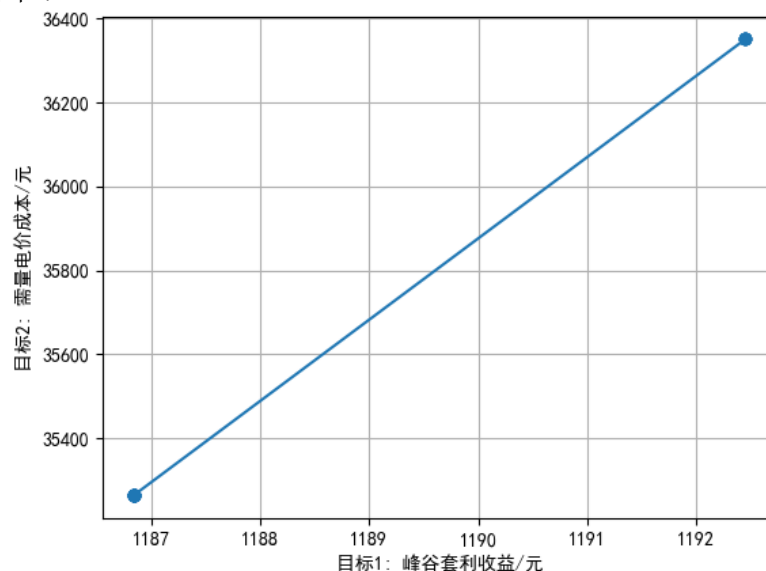


图4-7真实数据三并柜系统帕累托前沿解集图

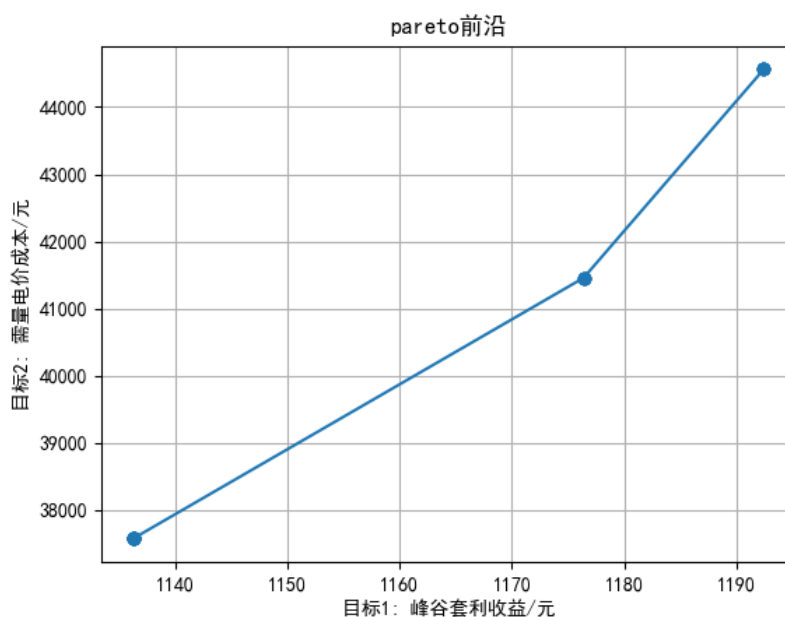


图4-8预测数据三并柜系统帕累托前沿解集图

对比图4-7和图4-8，相比于前两个，三并柜的预测结果准确度要差一点，但是也在差不多的范围。

2、基于德尔菲法选择最终方案

采用德尔菲法选择的单柜储能系统的优化策略，三并柜的最终方案收益见表4-4，储能策略见图4-9，具体值见附件。

表4- 4三并柜储能策略收益表			
数据	峰谷套利收益/元	需量电价/元	合同最大需量/KW
真实数据	1186.84	35263.47	734.66
预测数据	1192.45	44563.84	928.41

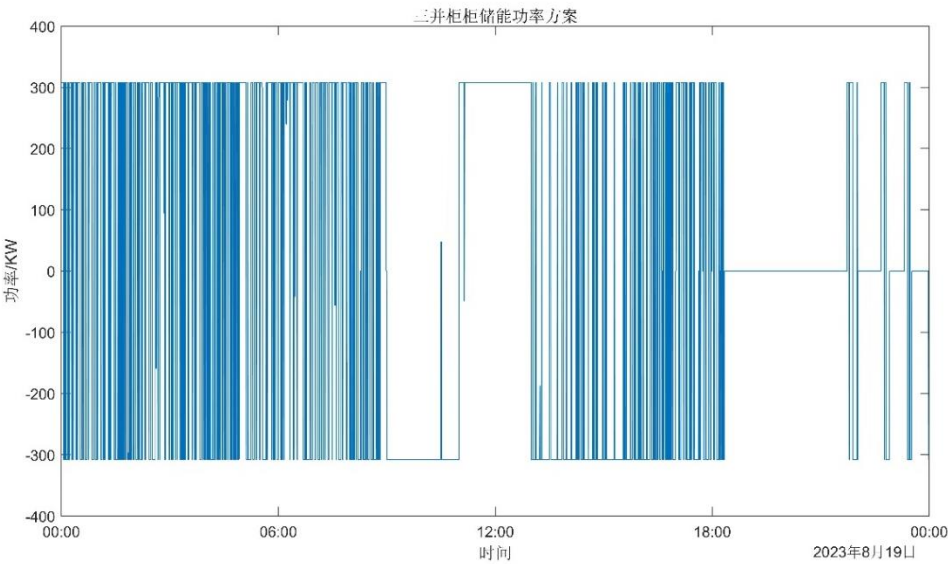


图4- 9三并柜储能设备优化策略图

与原策略相比，当日的峰谷套利收益增加244.4元，月需量电费减少1763.17元。

五、问题三模型建立与求解

5.1 问题分析

问题三要求在假定最大需量不受限制的情况下重新考虑储能设备的充放电策略，从而尽可能获得最大收益，并与上一问的充放电策略进行比较。首先在第二问所建模型的基础上进行修改，将最大需量不超过合同最大需量的105%这一约束删除，然后修改需量电费计算公式，加上超过合同最大需量的105%的电费，得到新的多目标模型。然后，求出每分钟储能系统的充放电策略和合同最大需量，并与原方案的收益进行比较。

5.2 模型建立

本题的决策变量与约束条件1、3、4、5均与问题二模型相同，仅需修改目标函数与约束条件2便可得到本题的多目标线性模型。

5.2.1 目标函数

$$\begin{aligned} \text{Max obj1} &= R \\ \text{Max obj2} &= -[H \times D + (N - H \times 1.05) \times D \times 2] \end{aligned} \quad (4-2)$$

其中 D 表示基础需量电价， $H \times D$ 表示未超过合同最大需量的105%时的当月需量电费，当超过时，超过部分的费用按基础价格的两倍计算，即公式（4-2）中的后半部分。

5.2.2 最大需量约束

1. 实际最大需量不小于合同最大需量的105%:

$$N - H \times 1.05 \geq 0 \quad (4-2)$$

这里直接不考虑实际需量小于合同最大需量的情况，因为当合同最大需量的105%超过实际最大需量时，实际峰谷套利收益不变，但是需量电费会大于两者相等时的电费，所以实际需量小于合同最大需量的情况不会出现。

5.2.3 整体模型

$$\begin{aligned} &\text{Max } R \\ &\text{Max } -[H \times D + (N - H \times 1.05) \times D \times 2] \\ &s.t. \begin{cases} \sum_{i=1}^j x_i \geq 0; i \in T \\ \frac{1}{60} \sum_{i=1}^j x_i \leq V; i \in T \\ s_i - x_i = h_i; i \in T \\ \frac{1}{15} \sum_{i=j}^{j+14} s_i \leq N; i \in [1, 1426] \\ N - H \times 1.05 \geq 0 \\ N \geq N_p \\ -U \leq x_i \leq U; i \in T \\ R = -\frac{1}{60} \sum_{i \in T} P_i \times x_i \\ x_i \in R; i \in T \\ s_i \geq 0, i \in T \\ H \geq 0 \\ N \geq 0 \\ R \geq 0 \end{cases} \end{aligned}$$

5.3 问题求解

本题求解思路与第二题类似，只需对具体模型进行少量调整，然后将各个储能系统的具体数据带入多目标线性模型，基于Gurobi求解器求得的帕累托前沿解集，并根据德尔菲法选择最终策略，各储能设备的优化策略见图5-1，其优化策略收益见表5-1，具体方案见附件。

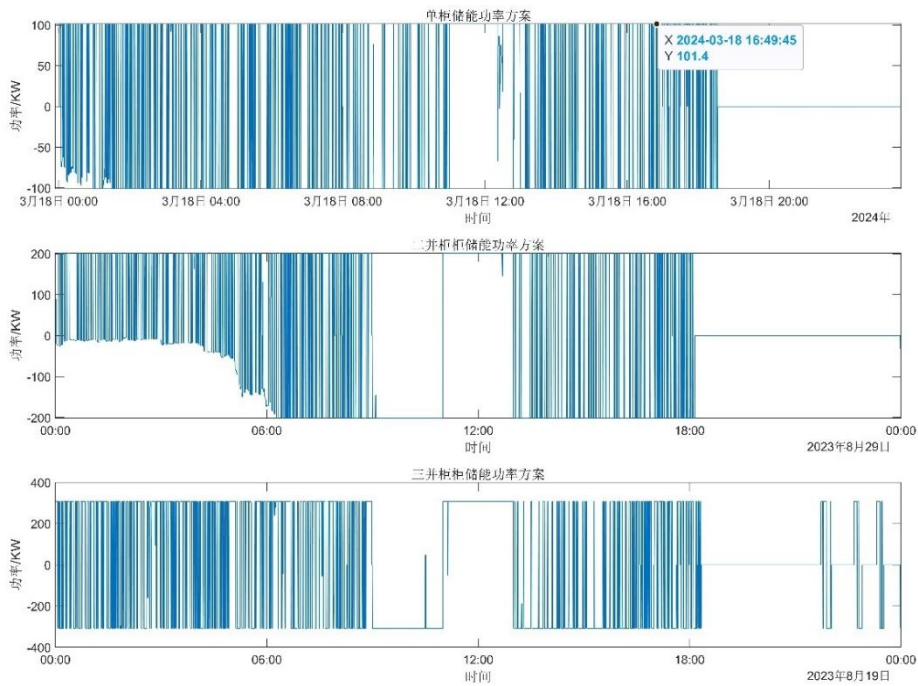


图5- 1各储能设备优化策略图

表5- 1各储能设备优化策略收益表

储能设备	峰谷套利收益/元	需量电价成本/元	合同最大需量/KW	是否超过合同最大需量的105%
单柜	332.07	25672.96	534.85	否
二并柜	785.81	33317.87	694.12	否
三并柜	1186.84	35263.47	734.66	否

与原策略相比其收益变化见表5-2：

表5- 2各储能设备收益变化表

储能设备	峰谷套利收益/元	需量电价成本/元
单柜	54.97	-868.16
二并柜	159.2	-1665.89
三并柜	244.4	-1763.17

将以上方案与问题二方案进行对比发现，问题三的结果同问题二相同。通过分析发现其原因为当实际最大需量超过合同最大需量的105%时，会导致需量电价的过度增加，进而降低总收益。

六、模型评价

6.1 模型的优点

(1) 线性规划模型：能够快速找到最优解，从而提高决策效率，可以在满足约束条件的前提下，实现资源的最优分配，可为合理利用有限人力、物力、财力等资源做出最优决策,提供科学依据。

(2) 多目标模型：能够同时优化多个目标对于某些多个具有相关性的任务，多任务模型可以实现任务间的信息交融，从而实现互相的提高，并且用一个模型实现多个任务，可以减少模型布置的数量和线上的资源消耗。

6.2 模型的缺点

(1) 线性规划模型：只能处理线性关系，对于数据的准确性要求高，只能对线性的问题进行规划约束，而且计算量大。

(2) 多目标模型：各目标加权值的分配带有较大的主观性，各目标之间通过决策变量相互制约，往往存在相互矛盾的目标，致使加权目标函数的结构十分复杂。

6.3 模型的推广与改进

(1) 动态优化模型：引入动态优化或实时调整机制，使模型能够根据实际情况和即时数据进行调整，从而进一步提高效率和适应性。

(2) 多目标优化：在在尽可能降低需量电价成本的前提下最大化峰谷套利收益的同时，可以引入其他优化目标，如最大化利用储能系统容量、减少充放电变化次数等，通过多目标优化模型，综合考虑多个因素，提高模型的实际应用价值。

(3) 改进的启发式算法：可以研究和应用改进的启发式算法（如遗传算法、模拟退火算法等），在求解数据量更大的问题时，能够在较短时间内找到接近最优解的解决方案。

参考文献

- [1] 韩文轩.两部制电价研究（上）：问题产生与现状[J].能源,2021,(05):24-29.
- [2] 任曦骏,宋竹萌,王宝,等.输配电价改革背景下两部制电价的应用现状与发展前景[J/OL].上海交通大学学报,1-18[2024-08-17].<https://doi.org/10.16183/j.cnki.jsjtu.2023.102>.
- [3] zgür zpeynirci, Kksalan M M .An Exact Algorithm for Finding Extreme Supported Nondominated Points of Multiobjective Mixed Integer Programs[J].Management Science, 2010.
- [4] Huang, Y., et al., Application of a hybrid model based on the Prophet model, I CEEMDAN and multi-model optimization error correction in metal price prediction. Resources Policy, 2022. 79: p. 102969.
- [5] 伍玉林,文福拴,丁剑鹰,洪晖虹,李承军.基于两部制电价的发电权交易模式[J].华北电力大学学报（自然科学版）,2010,37(5):16-22
- [6] 杨益,彭涛,许娟,陶勇,王艳龙,秦正阳.基于负荷曲线的最大需量估算方法研究[J].安徽电气工程职业技术学院学报,2023,28(4):33-38
- [7] Monnet, D.L., et al., Making sense of antimicrobial use and resistance surveillance data: application of ARIMA and transfer function models. Clinical Microbiology and Infection, 2001. 7: p. 29-36.

附录

代码1

介绍：第二题线性规划代码

```
def Gurobi_2_2_LP(P,FH,D_price,iop_upper,V,history_D,mobj,weight):  
    """  
    输入：  
    P: 电价  
    D_upper: 合同最大需量  
    FH: 负荷  
    iop_upper: 充放电功率上限  
    V: 储能系统容量  
    输出：  
    储放能方案  
    """  
  
    # 数据处理  
    n = 24*60  
    T = range(n)  
  
    # 创建模型  
    model = gp.Model('Q2_2_lp')  
  
    # 定义变量  
    # 第i分钟充放电功率  
    x = model.addVars(n,lb = -iop_upper, ub = iop_upper, vtype=GRB.CONTINUOUS, name='x')  
  
    # SD市电  
    SD = model.addVars(n, vtype=GRB.CONTINUOUS, name='SD')  
  
    # 需量  
    D = model.addVar(vtype=GRB.CONTINUOUS, name='D')  
  
    # 合同最大需量  
    D_upper = model.addVar(vtype=GRB.CONTINUOUS, name='D_upper')  
    # 峰谷套利收益  
    R = model.addVar(vtype=GRB.CONTINUOUS, name='R')  
  
    # 目标函数：最大化  
    if mobj:  
        model.setObjective(R, GRB.MAXIMIZE)
```



```

# model.setObjectiveN(-D_upper * D_price, index=1, priority=0, weight=0.003)
model.setObjectiveN(-D_upper * D_price, index=1, priority=0, weight=weight)
model.setParam(GRB.Param.PoolSolutions, 100)

else:
    model.setObjective(R - weight * D_upper * D_price, GRB.MAXIMIZE)

# 约束条件
# 储能容量
model.addConstrs(( quicksum(x[i] for i in range(j)) <= V * 60
                    for j in T),
                  name='V_upper')
model.addConstrs((quicksum(x[i] for i in range(j)) >= 0
                    for j in T),
                  name='V_lower')

# 满足负荷
model.addConstrs(SD[i] - x[i] == FH[i] for i in T)

# 需量电价约束
model.addConstrs(( quicksum(SD[i] for i in range(j,j+15)) <= D * 15
                    for j in range(n-14))
                  , name='D')
model.addConstr(D <= 1.05 * D_upper, name='D_upper')

# 历史需量约束
model.addConstr(D >= history_D, name='history_D')

# 峰谷套利收益
model.addConstr(R == quicksum((1/60) * P[i] * (-x[i]) for i in T), name='R')

## 原方案解
# model.addConstr(D_upper*D_price <= 26542, name='D_upper_0')
# model.addConstr(R>=277, name='R_0')

# 设置时间限制为7200秒，并行线程为12，不显示求解信息

```

```

# model.setParam('TimeLimit')
# model.setParam('Threads', 12)
model.setParam('OutputFlag', 0)
model.setParam('Threads', 12)
# 求解
model.optimize()

# 输出结果
optimil_value = 0
status = model.status
opt_x = None
opt_SD = None
opt_D_upper = None
opt_R = None
opt_Dbudget = None
opt_D = None

if status == GRB.OPTIMAL and not mobj:
    optimil_value = model.objVal
    opt_x = [x[i].x for i in T]
    opt_SD = [SD[i].x for i in T]
    opt_D_upper = D_upper.x
    opt_R = R.x
    opt_Dbudget = D_upper.x * D_price
    opt_D = D.x
elif status == GRB.OPTIMAL and mobj:
    nSolutions = model.SolCount
    if nSolutions > 10:
        nSolutions = 10
    print(f'Objective values for first {nSolutions} solutions:')
    for i in range(model.NumObj):
        model.setParam(GRB.Param.ObjNumber, i)
        objvals = []
        for e in range(nSolutions):
            model.setParam(GRB.Param.SolutionNumber, e)

            objvals.append(model.ObjNVal)
            opt_x = [x[i].x for i in T]
            opt_SD = [SD[i].x for i in T]
            opt_D_upper = D_upper.x
            opt_R = R.x
            opt_Dbudget = D_upper.x * D_price

```

<pre> opt_D = D.x print(f"\tSet{i}" + "".join(f" {objval:6g}" for objval in objvals[:3])) return status,optimil_value,opt_R,opt_Dbudget,opt_x,opt_SD,opt_D_upper,opt_D </pre>
代码2
介绍：第三题线性规划代码
<pre> def cla_Q2_3(P,FH,D_price,iop_upper,V,history_D,ws,name): # 文件名 pareto_solutions_name = 'q2_3_' + name + '_pareto_solutions.csv' projects_name = 'q2_3_' + name + '_projects.csv' fig_name = 'q2_3_' + name + '_pareto_front.png' w = 0.003 pareto_solutions = [] projects = [] for w in ws: # 求解 status,optimil_value,opt_R,opt_Dbudget,opt_x,opt_SD,opt_D_upper,opt_D,is_over = Gurobi_2_3_LP(P,FH,D_price,iop_upper,V,history_D,mobj = False,weight = w) pareto_solutions.append([w,opt_R,opt_Dbudget,opt_D,opt_D_upper,is_over]) projects.append(opt_x) # 矩阵转置projects,并保留4位小数 projects = np.array(projects).T projects = np.round(projects, 4) np.savetxt(pareto_solutions_name, pareto_solutions, delimiter = ',') np.savetxt(projects_name, projects, delimiter = ',') # 绘制帕累托前沿 plot_pareto(pareto_solutions,fig_name) # 求解函数 def cla_Q2_2(P,FH,D_price,iop_upper,V,history_D,ws,name): """ 输入： P实际电价 FH负荷 D_price需量电价 iop_upper合同最大需量 V储能容量 输出： """ # 文件名 </pre>

```

pareto_solutions_name = 'q2_2_' + name + '_pareto_solutions.csv'
projects_name = 'q2_2_' + name + '_projects.csv'
fig_name = 'q2_2_' + name + '_pareto_front.png'
w = 0.003
pareto_solutions = []
projects = []
for w in ws:
    # 求解
    status,optmil_value,opt_R,opt_Dbudget,opt_x,opt_SD,opt_D_upper,opt_D
Gurobi_2_2_LP(P,FH,D_price,iop_upper,V,history_D,mobj = False,weight = w)
    pareto_solutions.append([w,opt_R,opt_Dbudget,opt_D,opt_D_upper])
    projects.append(opt_x)
# 矩阵转置projects,并保留4位小数
projects = np.array(projects).T
projects = np.round(projects, 4)
np.savetxt(pareto_solutions_name, pareto_solutions, delimiter = ',')
np.savetxt(projects_name, projects, delimiter = ',')
# 绘制帕累托前沿
plot_pareto(pareto_solutions,fig_name)
if __name__ == '__main__':
    # 实际数据
    # 当月历史最大需量
    history_D1,history_D2,history_D3 = 552.9400,728.8283,771.3883
    # 电价数据
    index_24_3 = [2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1]
    value_24_3 = [1.1353, 0.688, 0.3096]
    price_24_3 = full_price(index_24_3,value_24_3)
    index_23_8 = [2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 2, 2, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2]
    value_24_3 = [1.2609, 1.0732, 0.3167]
    price_23_8 = full_price(index_23_8,value_24_3)

    # 电柜功率数据
    iop_upper1 = 101.4
    iop_upper2 = 201.5
    iop_upper3 = 308.1

    # 储能容量
    V1 = 215
    V2 = 215 * 2
    V3 = 215 * 3

    # 需量电价

```

```
D_price1 = 48
```

```
D_price2 = 48
```

```
D_price3 = 48
```

```
# TODO: 负荷数据
```

```
FH1,FH2,FH3 = read_FH('FHs_real.csv')
```

代码3

介绍：问题二、三求解代码

```
def cla_Q2_3(P,FH,D_price,iop_upper,V,history_D,ws,name):
```

```
# 文件名
```

```
pareto_solutions_name = 'q2_3_' + name + '_pareto_solutions.csv'
```

```
projects_name = 'q2_3_' + name + '_projects.csv'
```

```
fig_name = 'q2_3_' + name + '_pareto_front.png'
```

```
w = 0.003
```

```
pareto_solutions = []
```

```
projects = []
```

```
for w in ws:
```

```
# 求解
```

```
status,optmil_value,opt_R,opt_Dbudget,opt_x,opt_SD,opt_D_upper,opt_D,is_over =  
Gurobi_2_3_LP(P,FH,D_price,iop_upper,V,history_D,mobj = False,weight = w)
```

```
pareto_solutions.append([w,opt_R,opt_Dbudget,opt_D,opt_D_upper,is_over])
```

```
projects.append(opt_x)
```

```
# 矩阵转置projects,并保留4位小数
```

```
projects = np.array(projects).T
```

```
projects = np.round(projects, 4)
```

```
np.savetxt(pareto_solutions_name, pareto_solutions, delimiter = ',')
```

```
np.savetxt(projects_name, projects, delimiter = ',')
```

```
# 绘制帕累托前沿
```

```
plot_pareto(pareto_solutions,fig_name)
```

```
# 求解函数
```

```
def cla_Q2_2(P,FH,D_price,iop_upper,V,history_D,ws,name):
```

```
'''
```

```
输入:
```

```
P实际电价
```

```
FH负荷
```

```
D_price需量电价
```

```
iop_upper合同最大需量
```

```
V储能容量
```

```
输出:
```

```
'''
```

```
# 文件名
```

```

pareto_solutions_name = 'q2_2_' + name + '_pareto_solutions.csv'
projects_name = 'q2_2_' + name + '_projects.csv'
fig_name = 'q2_2_' + name + '_pareto_front.png'
w = 0.003
pareto_solutions = []
projects = []
for w in ws:
    # 求解
    status,optmil_value,opt_R,opt_Dbudget,opt_x,opt_SD,opt_D_upper,opt_D
Gurobi_2_2_LP(P,FH,D_price,iop_upper,V,history_D,mobj = False,weight = w)

    pareto_solutions.append([w,opt_R,opt_Dbudget,opt_D,opt_D_upper])
    projects.append(opt_x)
# 矩阵转置projects,并保留4位小数
projects = np.array(projects).T
projects = np.round(projects, 4)
np.savetxt(pareto_solutions_name, pareto_solutions, delimiter = ',')
np.savetxt(projects_name, projects, delimiter = ',')

# 绘制帕累托前沿
plot_pareto(pareto_solutions,fig_name)

if __name__ == '__main__':
    # 实际数据
    # 当月历史最大需量
    history_D1,history_D2,history_D3 = 552.9400,728.8283,771.3883
    # 电价数据
    index_24_3 = [2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1]
    value_24_3 = [1.1353, 0.688, 0.3096]
    price_24_3 = full_price(index_24_3,value_24_3)
    index_23_8 = [2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 2, 2, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2]
    value_24_3 = [1.2609, 1.0732, 0.3167]
    price_23_8 = full_price(index_23_8,value_24_3)
    # 电柜功率数据
    iop_upper1 = 101.4
    iop_upper2 = 201.5
    iop_upper3 = 308.1
    # 储能容量
    V1 = 215
    V2 = 215 * 2
    V3 = 215 * 3

```

```
# 需量电价
D_price1 = 48
D_price2 = 48
D_price3 = 48
# TODO: 负荷数据
FH1,FH2,FH3 = read_FH('FHs_real.csv')
# 求解第二题
print('求解第二题:')
# 单柜
print('单柜:')
ws1 = ws2 = ws3 = [0.001*i for i in range(1,31)]
cla_Q2_2(price_24_3,FH1,D_price1,iop_upper1,V1,history_D1,ws1,'1')
# 双柜
print('双柜:')
cla_Q2_2(price_23_8,FH2,D_price2,iop_upper2,V2,history_D2,ws2,'2')
# 三柜
print('三柜:')
cla_Q2_2(price_23_8,FH3,D_price3,iop_upper3,V3,history_D3,ws3,'3')
print('第二题求解完成')
# 求解第三题
print('求解第三题:')
# 单柜
print('单柜:')
cla_Q2_3(price_24_3,FH1,D_price1,iop_upper1,V1,history_D1,ws1,'1')
# 双柜
print('双柜:')
cla_Q2_3(price_23_8,FH2,D_price2,iop_upper2,V2,history_D2,ws2,'2')
# 三柜
print('三柜:')
cla_Q2_3(price_23_8,FH3,D_price3,iop_upper3,V3,history_D3,ws3,'3')
print('第三题求解完成')
```

由于本文篇幅有限剩余的代码将在附件中给出
