关于太阳能发电量的预测报告

一、原理

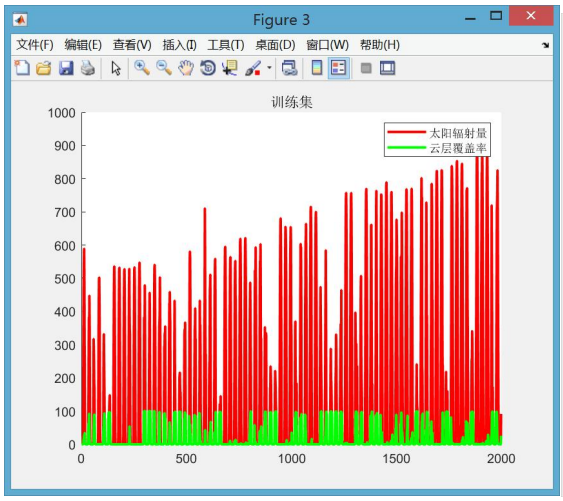
1、太阳能发电的周期性因素：我们知道太阳能发电是利用太阳在地表的辐射能量，转换成为电能，而太阳的辐射跟日地间的距离，以及地球上某地的地日角有关。日地距跟地球的公转有关，日地角跟地球的自转有关。不论是地球的自转还是公转都是一个周期量，有规律可循。所以我们想通过机器学习的方式发现这个背后的数据规律，比较准确的预测太阳能发电。

2、太阳能发电的干扰因素：我们在对干扰因素的逐一排查中，认为云层的覆盖率是影响太阳能发电的主要因素。云层可以吸收大量的太阳能辐射，导致到达地球表面的太阳辐射减少，所以我们在考虑太阳能发电视，必须考虑到云层覆盖率这个因素，方可帮助我们精准的预测太阳能的发电。

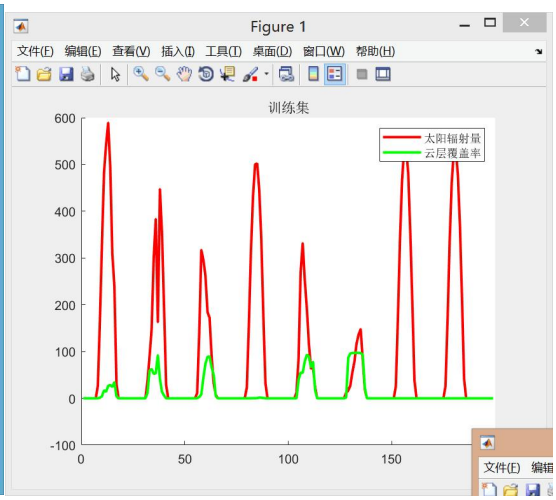
3、采取什么样的神经网络进行及其学习。综合了以上两点对太阳能发电的主要影响，我们可以得出这是一个周期性函数受干扰因素影响。基于这一点我们采用了 Nonlinear autoregressive neural network with external input。其中非线性自回归符合我们周期性函数的需求。而额外的输入参量有符合了我们对干扰因素的考虑，所以该神经网络符合我们的需求。

二、数据发掘

我们收集的大量的数据其中包括了温度、湿度、太阳能辐射，云层覆盖率等等。通过卷积的神经网络分析了太阳能发电和那个因素相关程度最高。实验证明太阳的发电量和云层的覆盖率，太阳能辐射相关度最高。于是我们进行数据分析。



2000 小时数据

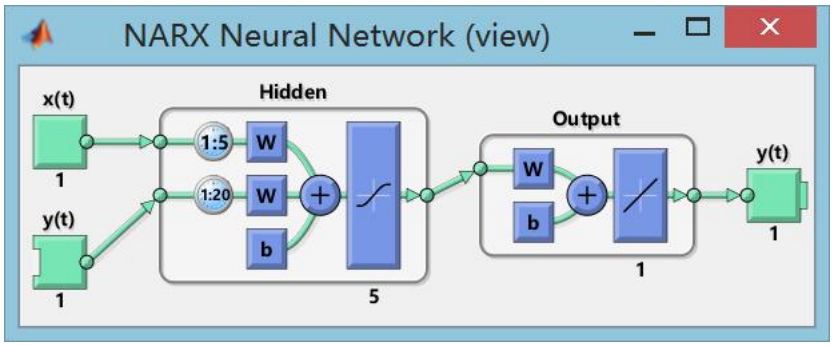


200 小时数据

红色数据是太阳能发电量，绿色为云层覆盖率，通过 2000 小时对的数据我们可以很容易的发现，红色数据呈上升的趋势，和绿色的数据成互补关系，通过 200 小时的数据我们可以更加直观的的发现他们两者之间的关系。一天之内，当某一个时间段云层覆盖率偏高的时候，太阳能发电就会变少，如果一阵的云层覆盖率居高不下那么太阳能发电将会整天减少。

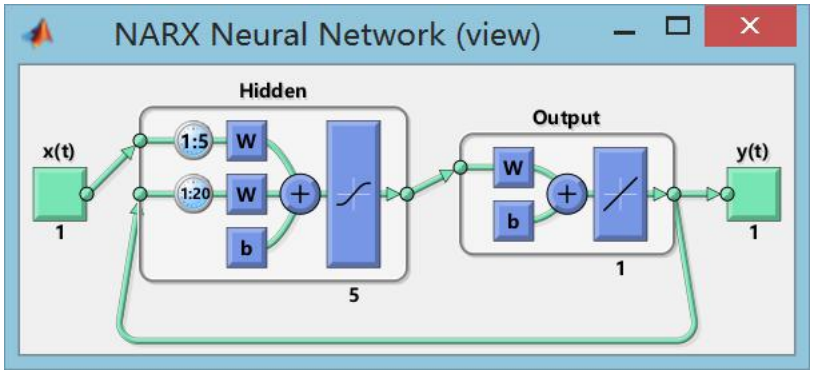
三、神经网络的学习

我们同构神经网路对以上的数据进行深度学习，在训练的时候我们采取如下的模型



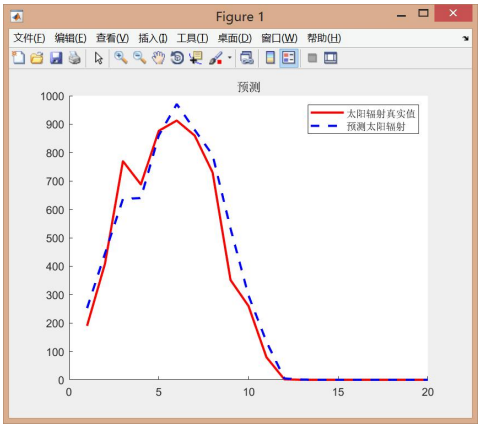
x 因素为云层的覆盖率，y 因素为太阳发电，因为我们其中历史太阳能发电为最为重要的因素，并且数据规律更加的复杂，所以我们采用了 20 个突触权值，云层覆盖率我们采用了 5 个突触权值（以上参数是经验所得），同时为了使结果更加的准确，我们采用 5 层隐层。同时因为是非线性所以加上了 sigmod 传递函数。最后经过输出层输出结果。

训练完毕神经网络之后，我们可以进行预测。通过历史的太阳能发电量，加以云层覆盖率，我们可以比较准确的预测太阳能发电，神经网络模型如下：



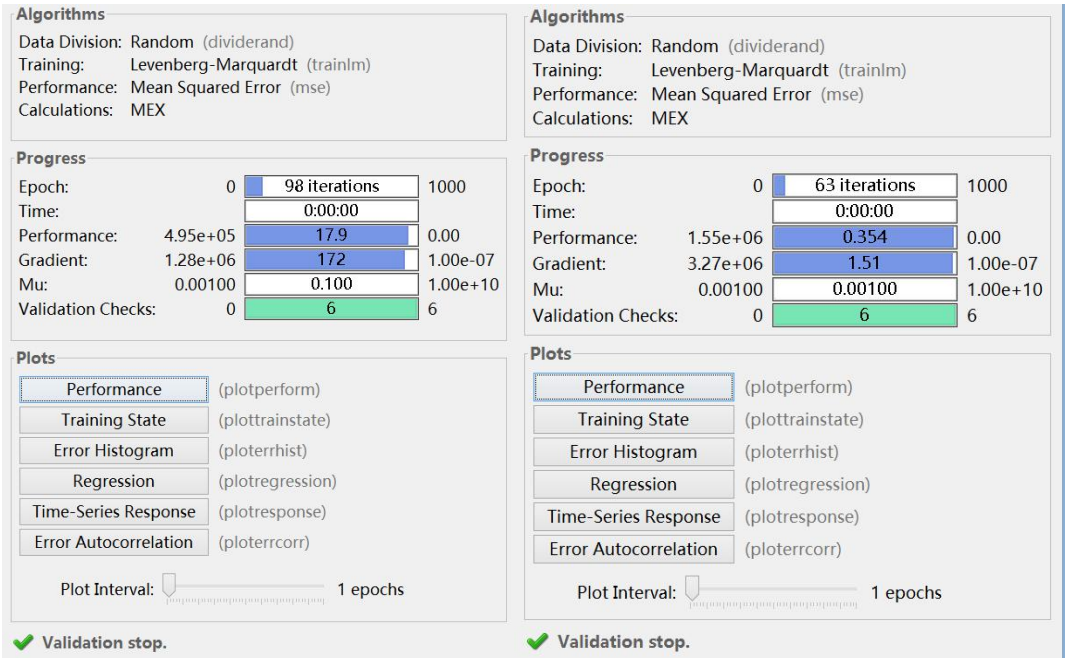
通过模型清楚地体现了回归性，每预测一个新的发电量，我们基于前十个发电量数据和干扰因素云层覆盖率，得到的新的结果又可以作为之后预测的依据。

四：预测结果：



未来 20 天预测值和真实值比较

预测的结果基本符合实际，保证了晚上太阳能发电量渐进为 0，中午是发电量较高，当有较厚的云层时，发电量下降，预测结果较好。



神经网络性能　　神经网络性能

Mgimss 调度算法

一、概述

当我们拥有了对太阳能发电的预测数据之后，唯一剩下的不确定的因素即为，未来会来多少任务，所以我们会在每一次新增任务的时候进行一次重新的规划。那么影响调度算法的因素有哪些呢？我们考虑到了有不同时刻的电价，太阳能的发电量，正在进行的任务，正在等待的任务，未来可能加入的任务，蓄电池的容量，太阳能的供电流向用电器还是流向电网，买入和卖出一度电的价格。为了简化我们的我们调度算法，我们先建立以下的模型。

1、所有的新能源的发电，都先存储在蓄电池之中，一旦蓄电池满，即反馈给电网。

2、优先使用蓄电池中的电能，如果不够则使用电网的点。

3、我们粗化了时间间隔，通过计时器每隔一定时间进行一次数据交互，同步

4、用电的优先级高于供电的优先级，当发电器和用电器同时向蓄电池传送或索要电能时，先解决用电器需求，在处理供电器需求。

该模型符合实际情况，又避免了新能源发电和用电器的直接交互，便于统一管理。

二、算法

我们采用了模拟最优解的算法。

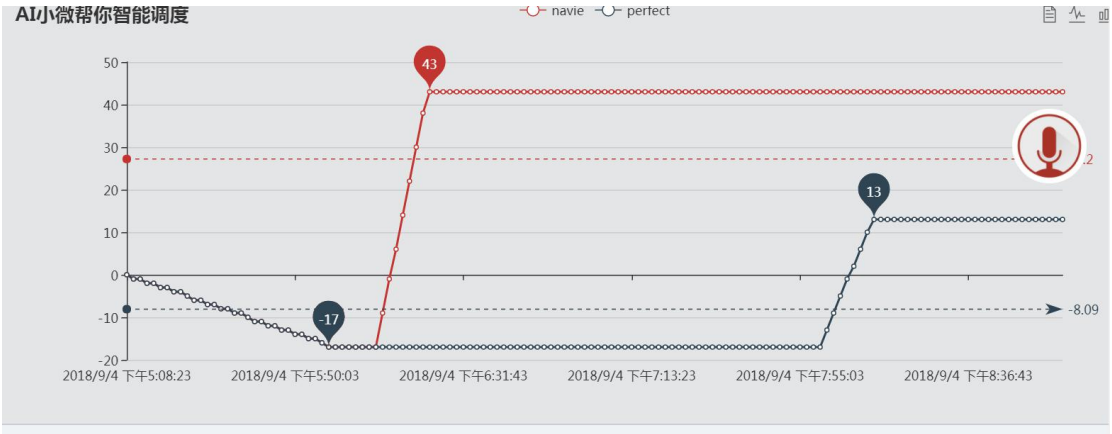
当我们执行调度算法时，我们知道有哪些任务正在执行，有那些任务等待执行，太阳能的预测数据，蓄电池的状态，电价等等因素。我们对每个等待执行的任务进行一一的判断，是否需要执行。

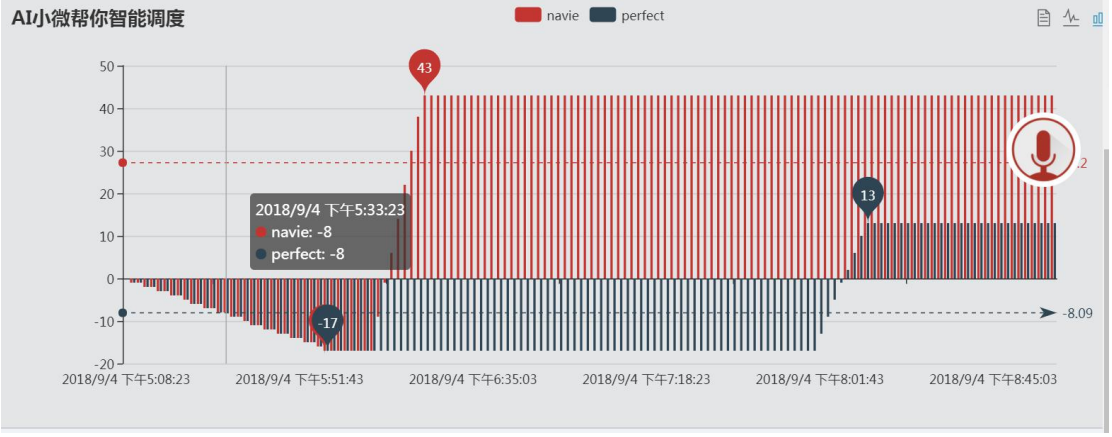
对每一个等待的任务，从这一时刻起到它必须完成的时刻前，如果所剩的时间不够等到下一次的调度，那么立刻将该任务加入到执行任务队列中。如果还有充足的时间，那么我们可执行的时间分成 N 份，比较该任务在这个 N 个时间段内开始执行所花费的价钱。取电价最少的时间段。再将其划分为 N 个时间区间，重复以上的比较，知道时间区间小于一指定值。如果最后当前的时刻落在最后的时间区间内，那么该等待的任务即进入执行任务的队列中。经过试验可得，用电器的开启往往在一些敏感时刻，比如电价下降的时刻，太阳能增加的时刻等等。这和我们的预期符合。

因为考虑到未来可能有用电器的加入，我们选择每个区间的起始点为该区间的代表，这样可以保证在太阳能利用率最高的情况下，尽早的完成该项任务，将多余的能量留给未来可能加入的用电器。

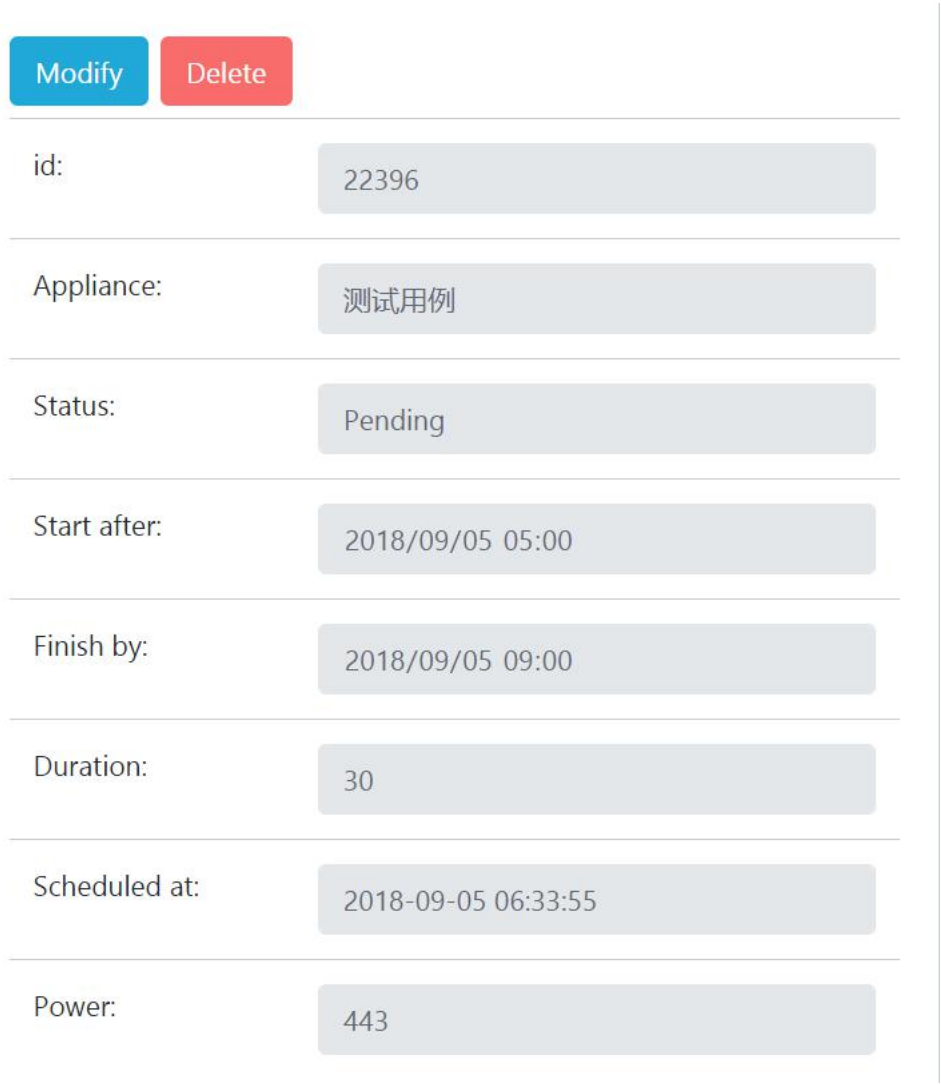
我们通过计算机很好的模拟真实的情况。基于此实现调度功能。

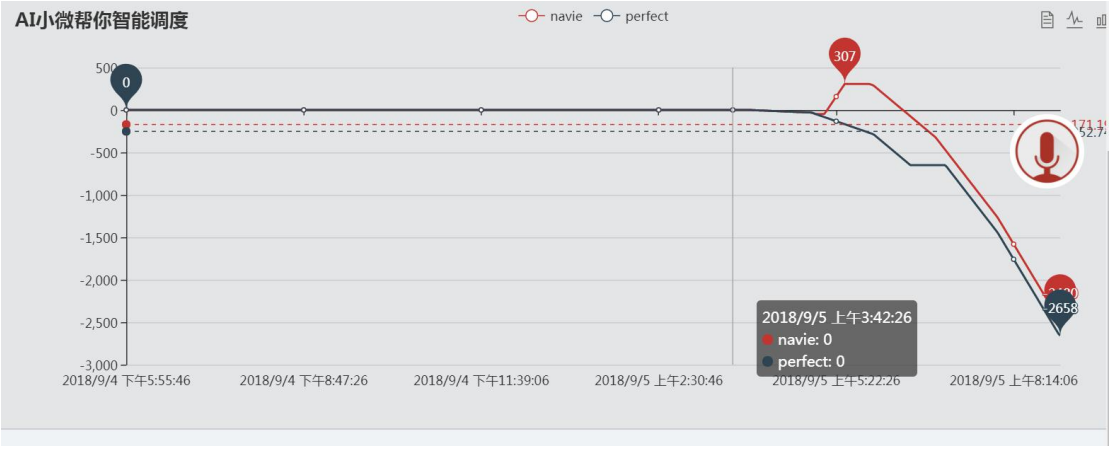
三、调度结果



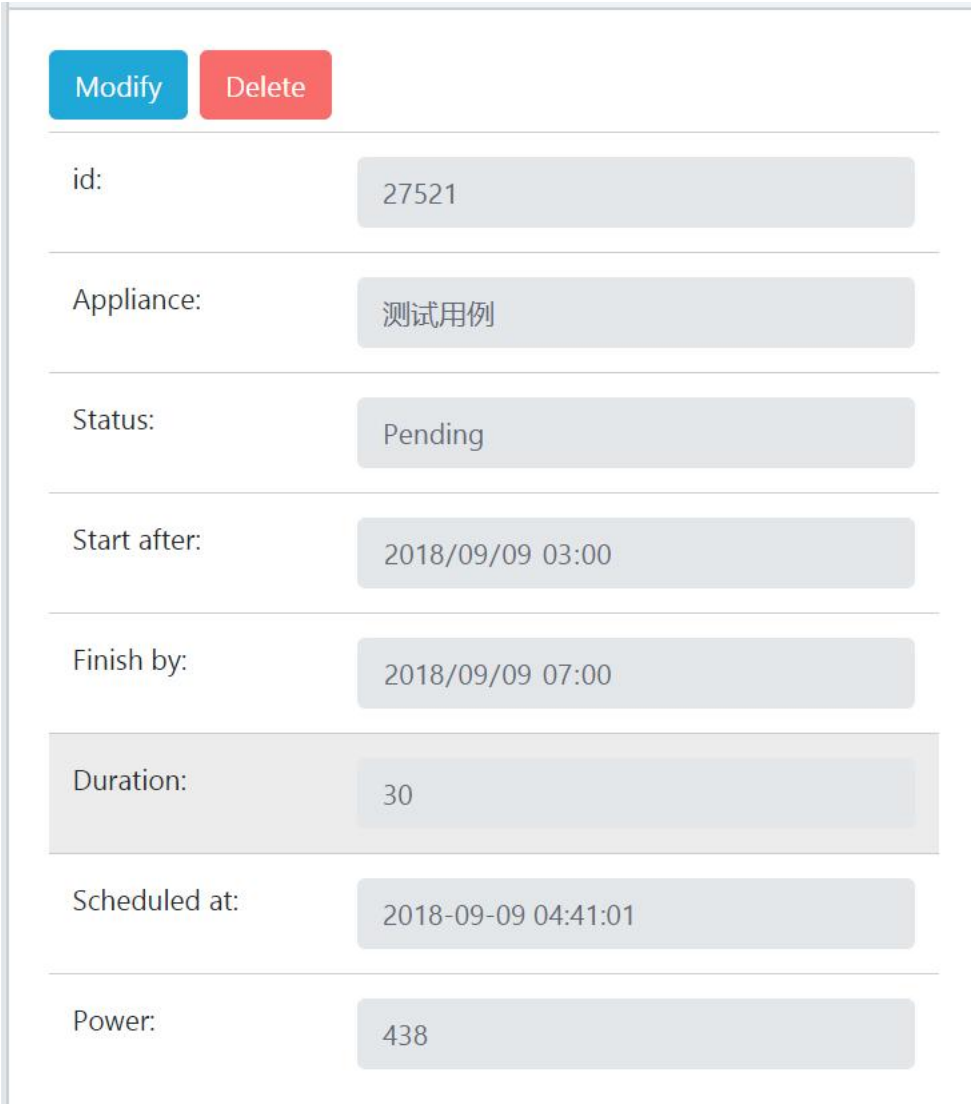


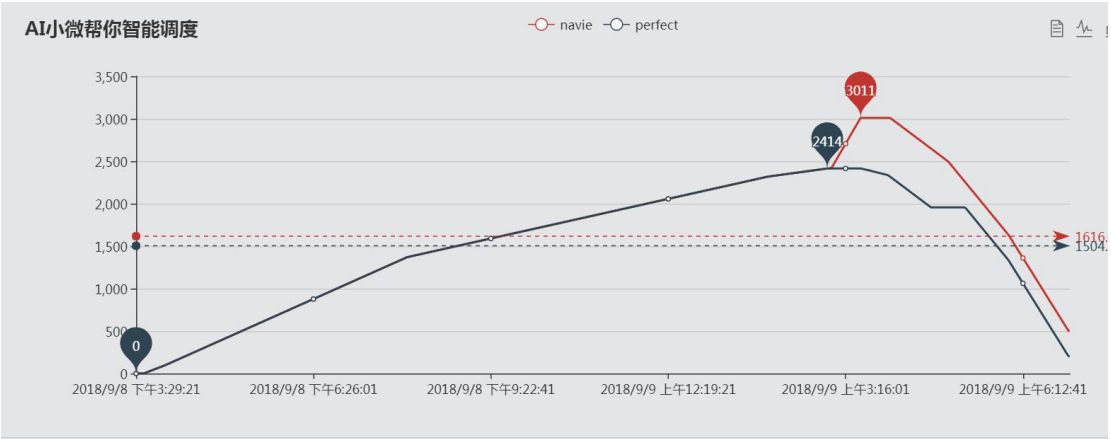
红色部分为 navie 的算法，黑色部分为我们通过调度算法结果。我们可以发现两者调度方式的结果相差巨大，虽然在晚上太阳能的预测渐进为零，但是由于站在晚上 8 点开始电价较低，花费的电费较少。



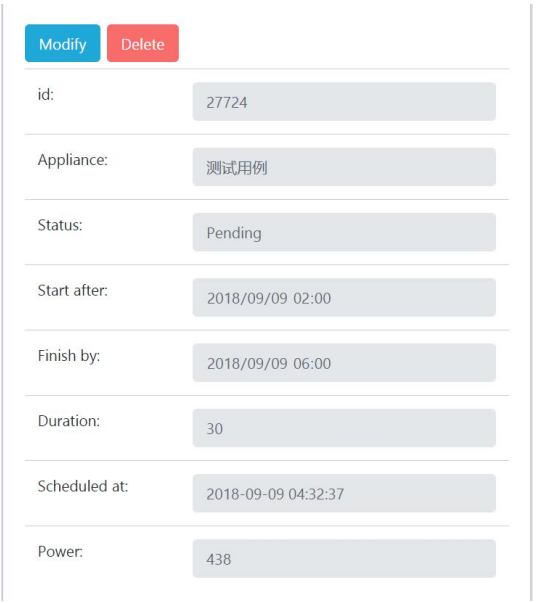


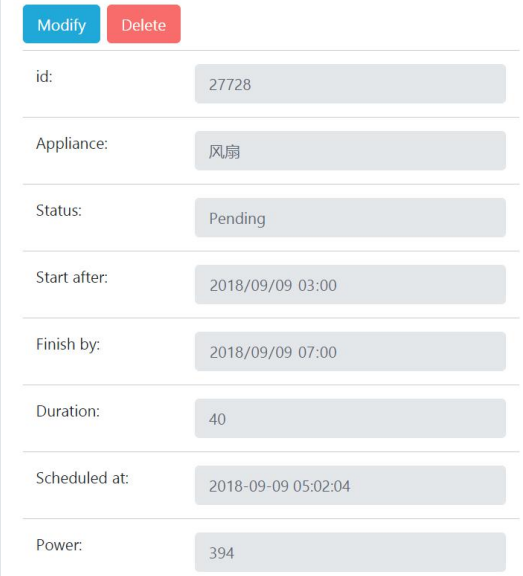
在该情况下，我们可以发现先早上五点开始有太阳能，并且逐渐增加，早上 8 点以后电价昂贵。我们课一发现调度算法选择了 6:33 分开始执行，虽然 7 点的发电量更为充足，但是因为太阳能电池中还储存着电能，所以并不一定需要等到 7 点才执行用电器，从 6:33 分开始（太阳能发电加上蓄电池原本的电量）既可以满足用电需求。这样可以将之后的太阳能电量留个未来可能需要用电的电器，实现太阳能电的高利用率。

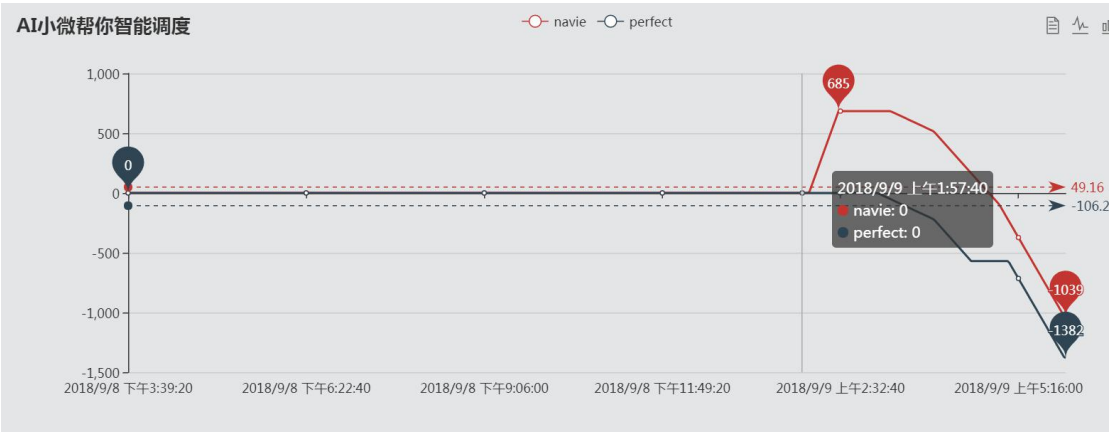


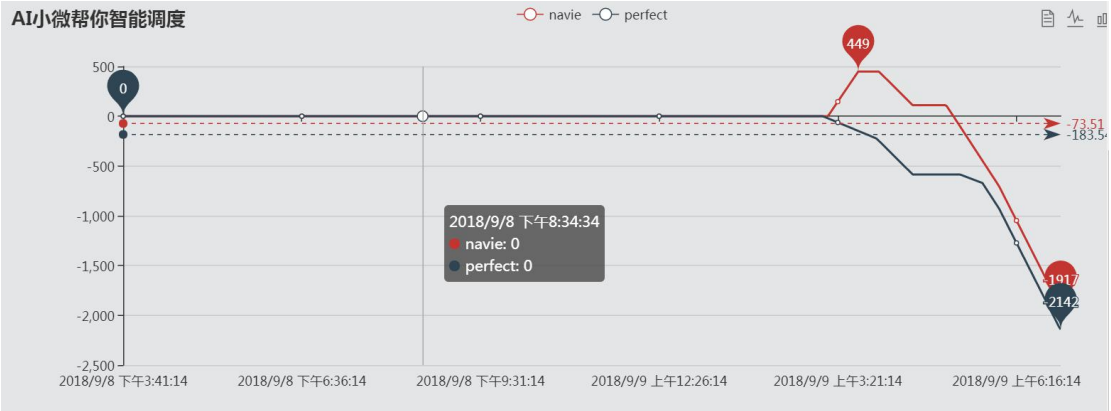


情况三：我们开启了两个用电器，一个为一直处于开启状态对的小灯泡，另一个为 3:00 到 7 点运行 30min 的测试用例，与情况二相比较，因为多了一个用电器，所以测试用例的开启时间往后推迟，从原本的 33 分推迟到了 41 分，推迟的 8 分钟就是为了弥补小灯泡所耗费的电能，两天曲线的趋势和情况二基本相同。按照调度算法所消耗电量小于 navie 的算法，符合预期。









在该情况下，我们同时调度了两个用电器，调度结果显示，两个用电器会在不同的时间区段执行，这样可以充分利用太阳能，另外由于 6 点之后的电价昂贵，两个用电器的额定功率不高，所以均安排在了 6 点之前电费较为便宜的时间段。该调度的情况说明，我们的调度算法可以为多个任务进行优化调度，调度结果符合预期。