Hi Tugba,

For the single charger optimizer you sent me before, I observe some interesting results and I would appreciate your insights. The issues are as follows.

There are some sessions that my station-opt got extremely large prices pairs, while in your optimizer it looks pretty good. I checked your optimizer, and find the following points:

First, the problematic session information is:

curr\_time = 990

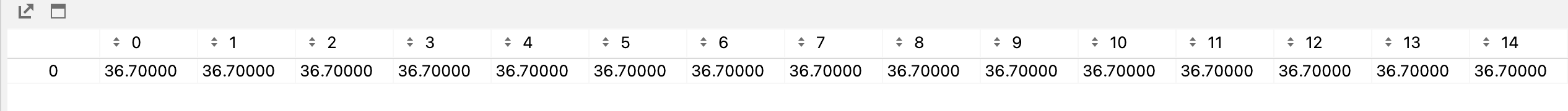
departure\_time = 1215

requested\_energy = 20

Currently user 8 is using

Here we use our previous minute time representation but I am sure **the real input time** unit is transformed in accordance with your input format.

And the TOU tariff here is:



which is **constant** for all time intervals, and **it is the extreme peak hour**.

Your optimizer will output:

curr\_time = 645

departure\_time = 1125

requested\_energy = 50

Currently user 6 is using

25 -0.011359560906853972

curr\_time = 645

departure\_time = 915

requested\_energy = 20

Currently user 7 is using

42 -1.4960284730136664e-05

**curr\_time = 990**

**departure\_time = 1215**

**requested\_energy = 20**

**Currently user 8 is using**

**1 -114.62210150016575**

The last line is the # of iterations, and the second term is improvement. For most of the sessions, as you can see, after dozens of iterations we converge with very little improvement(for example, 42 times, and -1.4960284730136664e-05).

However, for the problematic session, we only got 1 iteration and a very negative improvement, then we stop our iteration. Note that the initial value of the iteration is set as

**Uk\_flex = np.zeros(….),** so here a negative first-round improvement means that, the all zeros solution of uk\_flex is even better than meeting the constraints.

Although the output prices look normal(around 21cents / kwh), I think we cannot claim that it converges because of the huge improvement gap(-114.62210150016575, compared with previous -1.4960284730136664e-05).

I think the problem originated from the initial values here. For example, if I give an initial value of uk\_flex = np.ones(…) \* station\_pow\_max, which is 6.6, then after iterations, we will also get an extremely prices pair around 37cents / kwh.

I think this observation is quite interesting and I have no idea which implementation is correct/better. I wonder if you have any insights on that.

Looking forward to your reply!

Best,

Yifei