Tutorial 2: Modelling Transmission

In our previous example the load and generation were at the same bus. In this tutorial we will see how to model the transmission of power from one bus to another.

The Branch component

A branch transmits power from one bus to another. A branch may be a transmission circuit or cable connecting buses at different substations, or a transformer connecting buses that are at different voltages in the same substation.

Adding a Branch

Clear the previous model by tapping the  icon on the Build toolbar.

Build the model shown in Figure 16 by adding two buses (tap the Bus button twice), then add a generator and a load (tap the Gen button and then the Load button), finally add a branch (by tapping the Branch button).

On the Main toolbar tap the Solve button, select all the solve settings to OFF, then tap the Solve Now button.

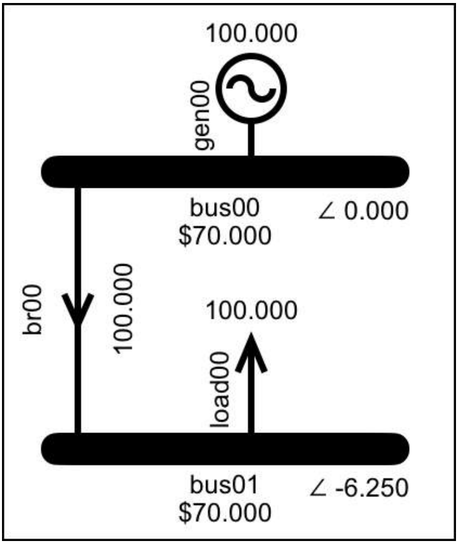

Figure 16: An electricity market model with transmission

Modelling a Branch

As discussed in Tutorial 1: Explaining Prices, the constraints of the LP model enforce the physical reality of the system being modelled. The physical reality of power flowing from one bus to another is enforced by two constraints; the power flow constraint and the maximum flow constraint.

The Power Flow constraint

The power flowing in the transmission system is Alternating Current (AC) electricity. The physical reality of AC power flow is modelled by an AC power flow equation.

The power flow constraint in the LP model implements a *simplified* AC power flow equation as shown in Equation 6.

$$Flow_{Branch} = (Angle_{FromBus} - Angle_{ToBus}) \times Susceptance_{Branch}$$

Equation 6: Simplified AC power flow equation

The simplified AC power flow equation looks similar in *form* to a DC power flow equation; hence a model that implements the simplified AC power flow is sometimes referred to as a DC power flow model.

Bus Angle

In Equation 6 the angle term refers to the phase angle of the voltage at the bus. In an AC system the voltage and current rise and fall, this happens 50 times per second in a 50Hz system. The voltage and current do not necessarily rise and fall at the same time. The voltage phase angle represents the timing difference between the rise and fall of the voltage and the rise and fall of the current.

How much AC power flows through a branch is proportional to the difference between the magnitude and phase angle of the voltage at one end of the branch and the magnitude and phase angle of the voltage at the other end.

In the simplified AC power flow, only the difference in the phase angle is used in the calculation; the magnitude of the voltage is taken to be the same at each end.

Because it is only the *difference* in phase angle that is used, it is only the relative phase angle of the buses that is important. Hence, in the LP model the phase angle at one of the buses is set to zero, this is the reference bus, and the solver is free to adjust all other phase angles relative to this.

The phase angle values determine how much power is scheduled to flow through the branches.

As shown in the Figure 16 result, after the solution is complete each bus has an associated phase angle. In this model bus00 is the reference bus for the electrical island. The first bus that is added becomes the reference bus, but the reference bus can be changed by selecting another bus as the reference bus, via its Data Display. Try changing the reference bus and re-solving to confirm that the phase angles are swapped around but the branch flow remains the same.

Branch Susceptance

The *impedance* of the branch impedes the flow of AC power. In the simplified AC power flow model, the impedance is represented by branch susceptance. Susceptance is related to the *inverse* of the impedance; hence susceptance is a multiplier not a divisor in Equation 6.

Equation 7 shows how susceptance is calculated from the branch's resistance R and reactance X.

$$Susceptance = \frac{-X}{R^2 + X^2}$$

Equation 7: Calculation of susceptance

The Maximum Flow constraint

The other branch constraint in this model is the maximum flow constraint shown in Equation 8.

$$Flow_{Branch} \leq MaxFlow_{Branch}$$

Equation 8: Branch Maximum Flow Constraint

This constraint ensures that the LP model does not allow more power to flow through the branch than the branch is capable of safely transmitting. We will see the impact of this constraint in the Binding Branch section.

The Node Balance constraint

As discussed in the single bus example, the node balance constraint ensures that the power that flows into a bus is equal to the power that flows out. In the single bus example this ensured that the cleared bid quantity was equal to the cleared offer quantity.

VARIABLES FOR BUS00

CONSTRAINTS FOR BUS00

```
bus00:
NodeBalance(LTE) constraint:
Shadow Price: $0.00
 -1.00000*br00_{BrFlowPos}
+1.00000*br00_{BrFlowNeg}
+1.00000*bus00_gen00_offer00_{Cleared}  <=
0.00000
```

```
bus00:
NodeBalance(GTE) constraint:
Shadow Price: $70.00
 +1.00000*br00_{BrFlowPos}
-1.00000*br00_{BrFlowNeg}
-1.00000*bus00_gen00_offer00_{Cleared}  <=
0.00000
```

Figure 17: Node balance constraints including branch flow

Now that the model includes transmission, i.e., a branch, the node balance constraints include branch flow, as shown in Figure 17, which is the Variables and Constraints display for bus00.

Branch Data Display

The branch parameters are viewed and edited via the branch's Data Display, shown in Figure 18.

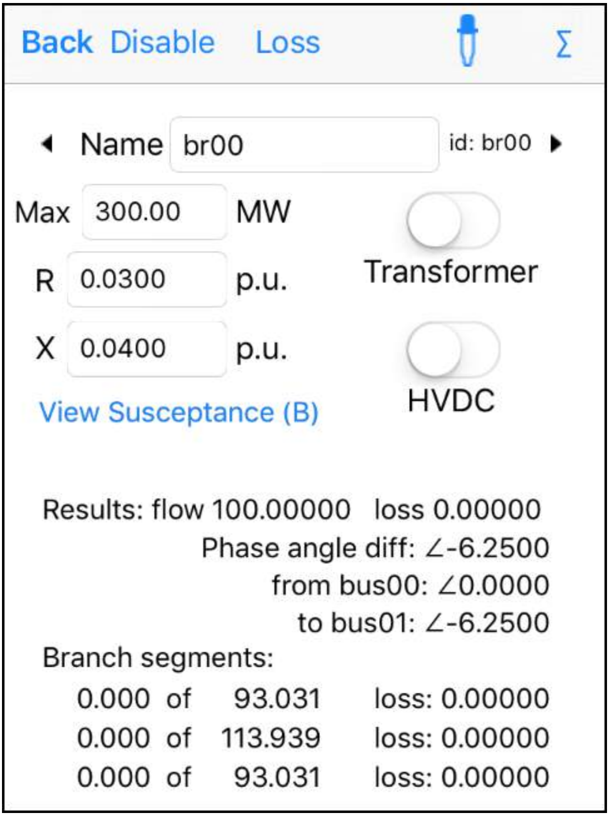

Figure 18: Branch Data Display

The power flow constraint uses the susceptance of the branch, calculated from the resistance and reactance. To view the susceptance, tap the "View Susceptance (B)" button and the display will change to show the calculated susceptance, see Figure 19.

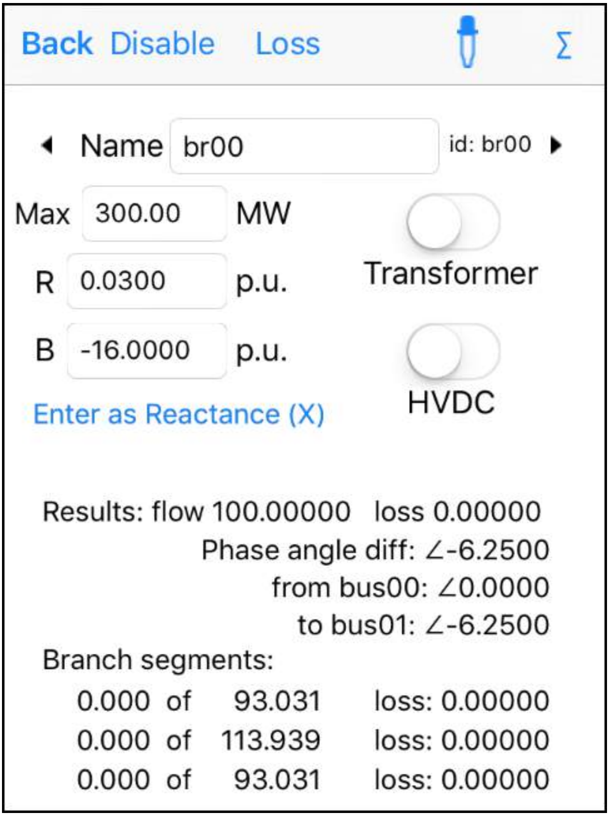

Figure 19: Branch Data Display, Susceptance option selected

Susceptance can be entered directly, in which case the reactance value will be set to zero. If a non-zero reactance value is entered, then susceptance will be re-calculated.

VARIABLES FOR BR00

```
br00_{BrFlowNeg}
non-basic
```

```
br00_{BrFlowPos}
100.000
```

CONSTRAINTS FOR BR00

```
br00:
PowerFlow(LTE) constraint:
Shadow Price: $0.00
 +1.00000*br00_{BrFlowPos}
-1.00000*br00_{BrFlowNeg}
+16.00000*bus01_{AnglePos}
-16.00000*bus01_{AngleNeg}  <= 0.00000
```

```
br00:
PowerFlow(GTE) constraint:
Shadow Price: $0.00
 -1.00000*br00_{BrFlowPos}
+1.00000*br00_{BrFlowNeg}
-16.00000*bus01_{AnglePos}
+16.00000*bus01_{AngleNeg}  <= 0.00000
```

```
br00:
BranchFlowMax(LTE) constraint:
Shadow Price: $0.00
 +1.00000*br00_{BrFlowPos}  <= 300.00000
```

```
br00:
BranchFlowMax(LTE) constraint:
Shadow Price: $0.00
 +1.00000*br00_{BrFlowNeg}  <= 300.00000
```

Figure 20: Variables and Constraints display for a branch

Branch variables and constraints

To view the branch's variables and constraints tap the ∑ button on the branch Data Display. This will present the Variables and Constraints display as shown in Figure 20, where you can see the implementation of the power flow and branch maximum flow constraints.

"From bus" and "To bus"

Positive branch flow is defined as being from the bus that has the lower alphabetical name, the "from bus", to the bus that has the higher alphabetical name, the "to bus".

Unrestricted variables

In the variables section of Figure 20 you will notice that there are two branch flow variables; a positive flow and a negative flow. This is because the simplex algorithm restricts all variables to have only positive values (the complete workings of the algorithm are covered in Tutorial 9: Simplex Explained).

If we only had one branch flow variable, we could only schedule branch flow in the positive direction. Hence, branch flow is modelled using two variables; one variable that represents flow in the positive direction and another that represents flow in the negative direction… both of these variables can only take positive values.

The displayed branch flow result is calculated from the positive and negative variables. The calculation takes place in post-processing after the simplex algorithm has solved the LP Model.

Non-basic variables

On the Variables and Constraints display a variable that is greyed out is non-basic. A non-basic variable has been set to zero by the simplex algorithm. This is because in order to find a feasible solution the simplex algorithm must set a certain number of variables to zero. The variables that are set to zero are the non-basic variables… they are not zero because a constraint limits them to zero, they are zero because the algorithm has determined that allowing them to be non-zero would not improve the result.

We can see from the result in Figure 20 that the reverse flow on the branch is a non-basic variable. The simplex algorithm did not find any way that allowing this variable to be non-zero could produce a better objective value.

For a full explanation of basic and non-basic variables see Tutorial 9: Simplex Explained.

Bus variables and constraints

When we looked at the variables and constraints in Tutorial 1: Explaining Prices, we saw that the bus did not have any associated variables. This was because the model did not contain any transmission, hence there were no power flow constraints and therefore no need for a phase angle variable.

Now that the model includes transmission, bus01 has a phase angle variable. Bus00 is the reference bus so its phase angle is set to zero, i.e., not a variable. The variables and constraints for bus00 and bus01 are as shown in Figure 21 and Figure 22.

VARIABLES FOR BUS00

CONSTRAINTS FOR BUS00

```
bus00:
NodeBalance(LTE) constraint:
Shadow Price: $0.00
 -1.00000*br00_{BrFlowPos}
+1.00000*br00_{BrFlowNeg}
+1.00000*bus00_gen00_offer00_{Cleared}  <=
0.00000
```

```
bus00:
NodeBalance(GTE) constraint:
Shadow Price: $70.00
 +1.00000*br00_{BrFlowPos}
-1.00000*br00_{BrFlowNeg}
-1.00000*bus00_gen00_offer00_{Cleared}  <=
0.00000
```

Figure 21: Variables and constraints for bus00

Figure 22: Variables and constraints for bus01

Note that bus01 has two phase angle variables. This is because the phase angle can be positive or negative, i.e., it is an un-restricted variable in the same way as the branch flow variable, and hence the same explanation applies… the simplex algorithm requires that all variables remain >=0, hence the phase angle is modelled as a positive angle variable and a negative angle variable, both of which can only take non-negative values.

Binding Branch

A branch is binding if its scheduled flow is equal to the limit set by its maximum flow constraint. To see the impact of a binding branch, lower the maximum flow on br00 to 40MW, i.e., less than the scheduled flow of 100MW.

To lower the branch's maximum flow, i.e., the branch limit, double tap branch br00 and lower its Max from 300MW to 40MW. Solve. The result is shown in Figure 23.

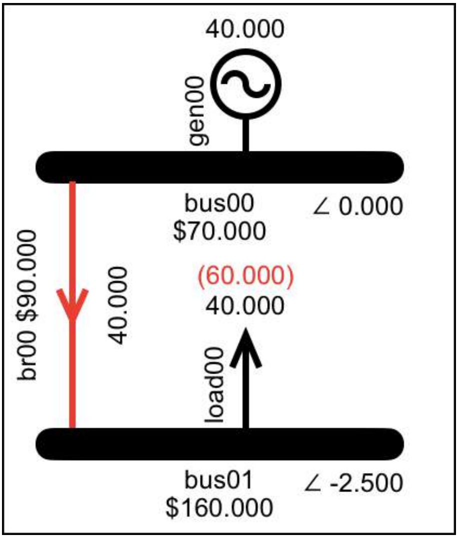

Figure 23: Binding branch

Price separation

Lowering the branch limit has produced a result where the simplex algorithm has not been able to fully clear the bids. This has led to price separation, whereby the prices at bus00 and bus01 are no longer related.

As explained in Tutorial 1: Explaining Prices, the bus price at bus00 is $70/MW due to the fact that relaxing the node balance constraint would allow the solver to reduce the existing $70/MWh generation offer, benefitting the objective value at a corresponding rate.

At bus01 things have changed due to the binding branch… relaxing the node balance constraint at bus01 would not allow the cleared generation to decrease. What it would do is allow more of the $160/MWh bid to clear, which would improve the objective at a rate of $160/MW; hence $160/MW is the price at bus01. Before the binding constraint was applied the $160/MWh bid was cleared to its limit, relaxing the node balance constraint would not allow it to clear any more, hence the $160/MWh price did not influence the bus price.

Minimizing price is not the objective

Note that the objective of the simplex algorithm is not to minimize the bus prices. Rather, the objective is to maximize the objective value, which will be achieved by maximizing the benefit of the cleared load bids while minimizing the cost of the cleared generation offers.

The prices are signals to indicate the potential for improving the objective value. Hence, in Figure 23 there is a higher price at bus01 than at bus00, because generation sited at bus01 would have a greater impact on the objective value.

Capacity Price

Looking at the result shown in Figure 23 we see that the branch is red, which indicates that the branch's maximum flow constraint is binding. Also, the branch has a price of $90 after its name. This $90/MW is the shadow price of the branch's maximum flow constraint.

Every constraint has a shadow price and these shadow prices are shown on the Constraints display. The shadow price of the branch maximum flow constraint only appears on the *main* display if it is non-zero, which only happens when the branch is binding. This non-zero shadow price indicates that relaxing the branch's maximum flow constraint will improve the objective value.

The shadow price of the br00 maximum flow constraint is $90/MW because an incremental increase in branch capacity would allow more generation to be cleared at bus00 and transmitted to bus01 where it would allow more bids to clear; the cleared bids would contribute another $160 of value to the objective, while the corresponding cleared generation offer would cost $70, for a net benefit of $90. You can confirm this by increasing the branch capacity by 1MW, viewing the corresponding change in scheduled quantities and observing on the Results display that the objective value increases by $90.

Congestion Charges

The financial impact of the binding branch is that the load pays more for purchasing electricity than the generation is paid for producing it. The total quantities are shown on the Results display in Figure 24. Here you can see that the load pays $6400, the generation is paid $2800. The $3600 difference is referred to as a congestion charge. This value is labelled $Grid on the display because it is a cost that arises due to the transmission network, also referred to as the grid.

Later when we model line losses, we will see that they can also give rise to the load paying more than the generation receives even when there are no binding branches. In this case the extra payments are referred to as transmission rentals.

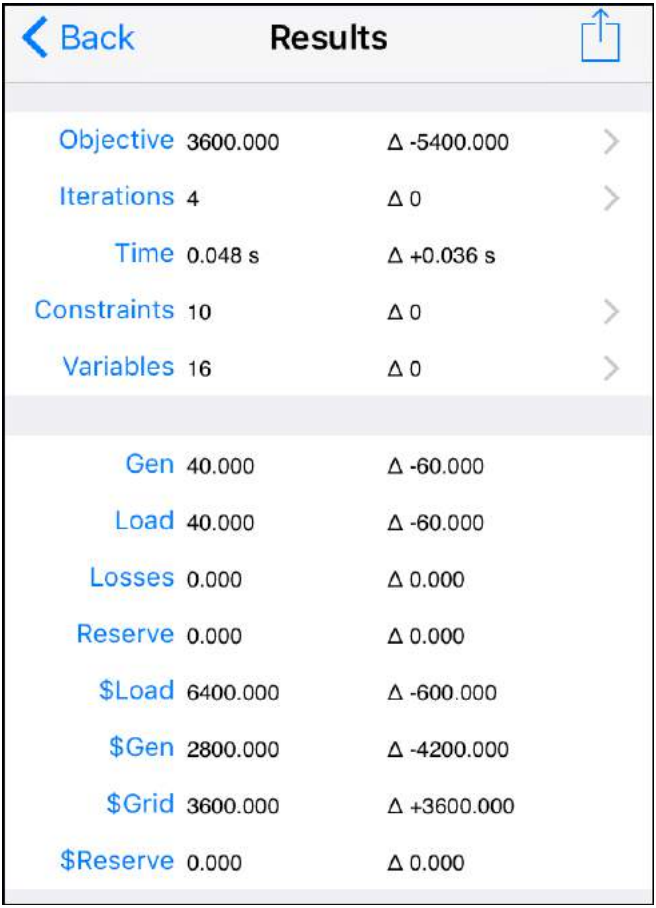

Figure 24: Binding branch leads to congestion charges, as indicated by the non-zero $Grid value

Shadow price and 1MW relaxation

We can also use this binding branch model to demonstrate that the shadow price indicates the per MW change to the objective value due to an *incremental* relaxation of the node balance constraint, which is not necessarily the value of the next 1 MW. Even though we have successfully been using 1MW to demonstrate how changes in the objective value relate to the shadow price, we are about to demonstrate that there are times when 1MW may be too large an increment.

To make this point, change the binding branch example by lowering the bid at load00 from 100MW to 40.5MW. If we solve now then the result is the same, the branch is still binding at 40MW.

To confirm the price at bus01, we are going to relax its node balance constraint by adding a generator with a 1MW offer and a $0 price to bus01 (similar to what we did in Tutorial 1: Explaining Prices). Tap the Gen button to add a generator to the model. Drag the new generator to bus01. Double tap the new generator and edit its offer block1 to be 1MW at $0. Solve, and the result is shown in Figure 25.

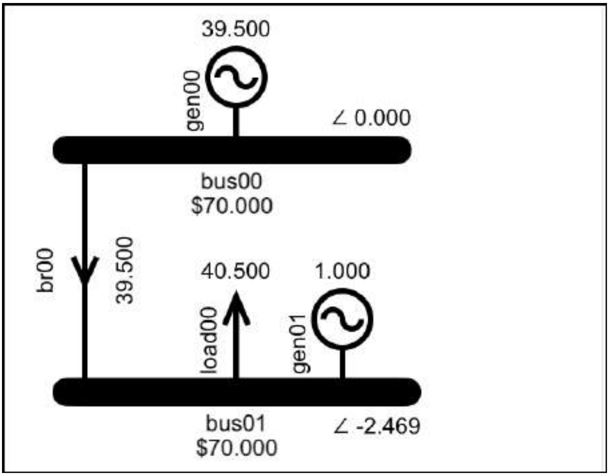

Figure 25: Bid at load00 changed to 40.5MW and gen01 added with an offer of 1MW at $0

The extra 1MW has resulted in the bid fully clearing and therefore the branch is no longer binding… the result has fundamentally changed, and we have *not* succeeded in proving the original $160/MW bus price. To prove the price, we need to relax only the constraint that is setting the bus01 price in the original result… that constraint was the node balance constraint at bus01, but by adding the 1MW of generation we have also relaxed the branch constraint.

Before we try again we need to return to the binding line result by setting the offered gen quantity at gen01 to 0MW then solving the model. Next, to relax just the node balance constraint and leave the branch binding, edit gen01 so that its offer quantity is 0.1MW at $0.

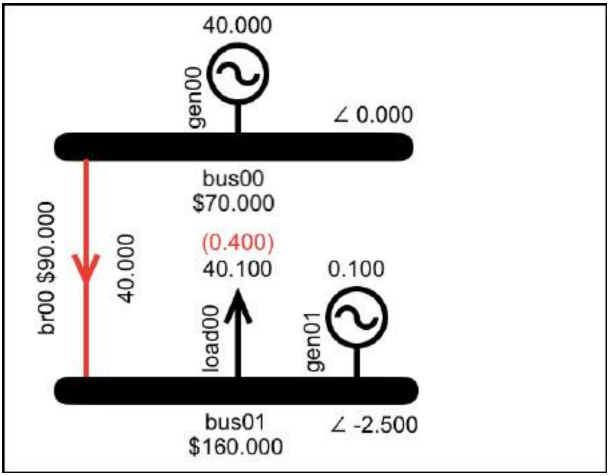

Figure 26: Bid 40.5MW and gen01 offering 0.1MW at $0

Solve again and the result is as shown in Figure 26. We still have the $160 bus price and we can see that the only change is that the bid has now cleared 40.1MW. Checking the Results display shown in Figure 27 we see that the 0.1MW increase in the cleared bid quantity has benefited the objective by $16. Dividing the benefit of $16 by the incremental quantity of 0.1MW we confirm that the per-MW value of generation at bus01 is $160/MW.

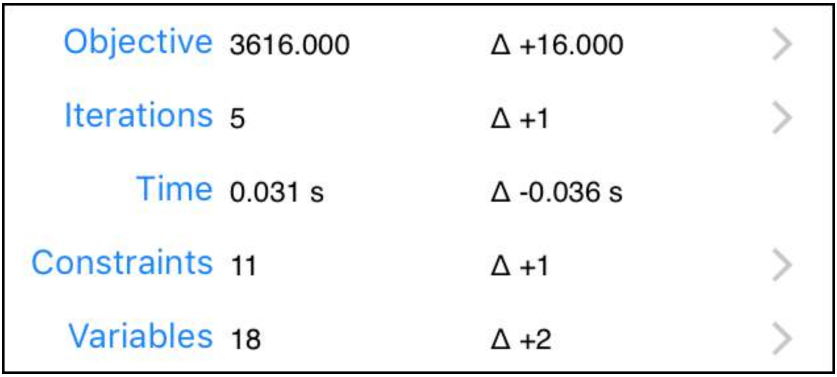

Figure 27: Adding 0.1MW at $0 results in objective value change of +$16, i.e., $160/MW

Summary

In this tutorial we added transmission to the model and saw how it is modelled by the power flow constraint, the branch limit constraint, and the inclusion of branch flow in the node balance constraints.

We saw how the power flow constraint leads to the solver using the phase angle variable to determine branch flow. We also saw that when the branch limit binds, the bus prices at either end of the branch become separated, i.e., a binding branch constraint causes price separation.

The binding branch example was also a useful opportunity to present an example showing that it is the value of an *incremental* change in the power available at the bus that determines the bus price, i.e., while it may be possible to use a 1MW change to illustrate the impact of an incremental change (as we have done earlier), it is also the case that 1MW may be too large to represent an *incremental* change.