<u>Department of Electrical Engineering</u>

<u>IIT Hyderabad</u>

<u>Power Storage Optimization of a Battery.</u>

This project is submitted as a part of Independent Project(EE2015) to partially fulfil the requirement of credits.

The faculty guide is Dr. Vaskar Sarkar, Associate Professor EE department has helped us and guided us throughout the project.

The project is done and submitted by –

Lakshit Singla - EE17BTECH11021

Pranjal Singh - EE17BTECH11031

Tarun Tej Reddy Ch.- EE17BTECH11042

Index: -

Introduction:

Every residential house or a commercial plant receives power primarily from a power station. But many times, the power provided by the power station isn’t sufficient. Hence some power has to be provided by external means.

We can supply this additional required power through a battery and also it can be borrowed from the grid to meet the energy requirements. The grid has different costs of power supply at different hours. The battery can also be charged from the grid simultaneously providing the load power to the house or the plant. Similarly, when there is enough power from the power station, the battery can supply power to the grid. In this way, we need to pay to the grid whenever we are taking power from the gird, and in turn, we can earn money by giving the extra power of the battery to the grid. It means, the battery can either be charge or discharge.

Our objective is to find out at what times do we need to take power from the grid (either for fulfilling load requirements or charging the battery or both together) and give power to the gird which will lead in the minimum cost paid to the grid.

Description of the Procedure:

We have to find the minimum price that we should spend on grid for a period of 24 hours. To find out this value, we give the values of pvpower at each hour, grid cost at each hour and load values at each hour in an excel file. We then use MATLAB to read this data and write this data into a gdx file. When we execute the MATLAB file, a gdx file is automatically created in the same folder with the name “matdata.gdx”. We use GAMS for the next part.

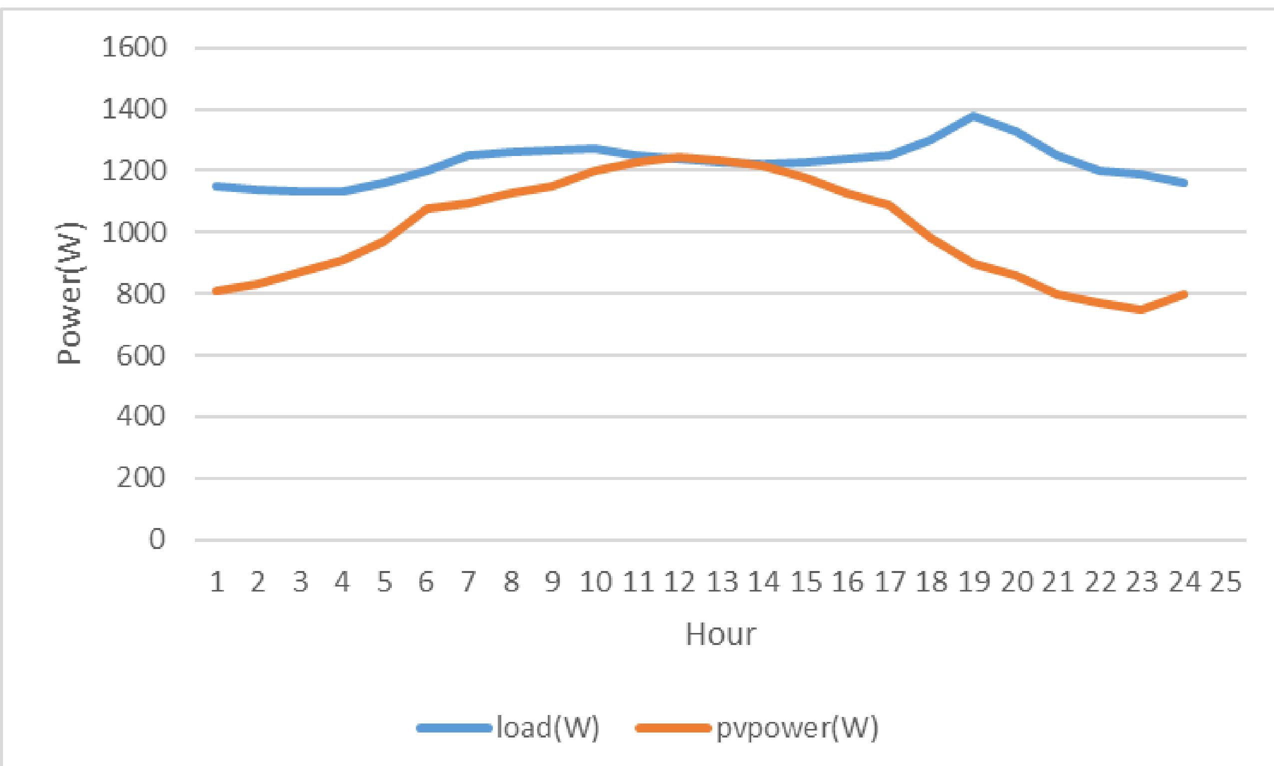
The General Algebraic Modelling System (GAMS) is a high-level modeling system for mathematical optimization. GAMS is designed for modeling and solving linear, nonlinear, and mixed-integer optimization problems. The data from the gdx file is imported to GAMS using a function. Then, we define sets and parameters for these inputs and write equations corresponding to the constraints and requirements. Using Mixed Integer Programming, we minimize our net cost paid to the grid for an entire 24-hour period.

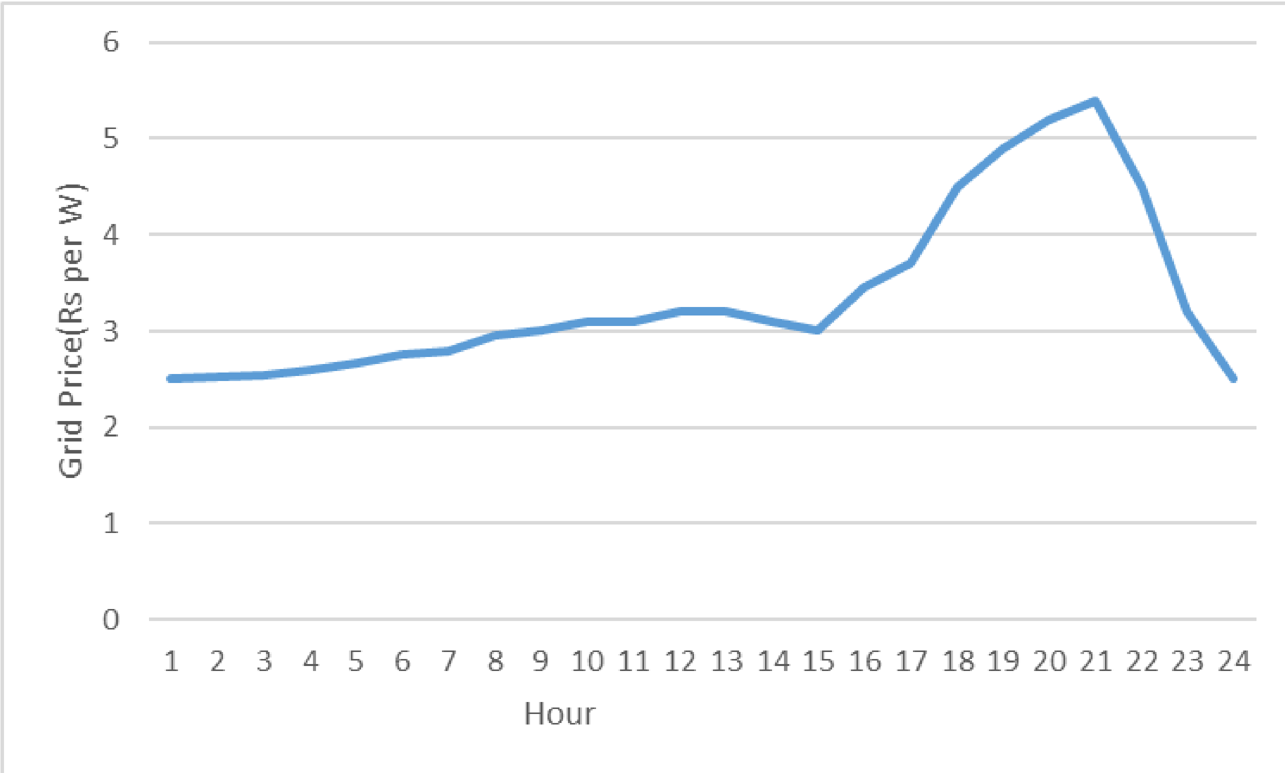
We send this result to display in MATLAB using “setout”. The minimum price along with the power taken from grid (or given to grid) is displayed in the MATLAB file.

Providing the Input values:

Input Data is stored in an excel sheet.

In the excel sheet, we provide the of the values of pvpower at each hour, grid cost at each hour and load values at each hour in the form of columns. These are the graphs of the inputs which are taken in the project file.





Exporting the data using MATLAB:

We have written a matlab code which reads the data from the excel sheet and then writes this data into a gdx file. This is achieved through a matlab function called xlsread. This command reads the data from the excel sheet and assigns the tables to a matlab variable which stores the table data. After getting the data, any manual calculation can be done on this data to achieve the desired values which would be used for optimization. After doing this, we need to convert the data into structures which can be passed on to the gdx file.

After this, we write the gams command and enter which data needs to be exported. We need to specify the gms file name to which it has to be exported.

Then execute the matlab code and we find that there is a matdata.gdx file created in the same directory of the matlab code. On viewing the gdx file, we can find the data in the form of columns either 1-d or 2-d depending on the data which we have sent. This gdx file is the interface between matlab and gams. It can be read and written by both the softwares. We have attached the matlab code in the project folder with the entire commands already written. Run the matlab code to proceed to optimization of price.

Loading the data to GAMS:

Open the matdata.gms file and enter the parameters which are needed to be imported to gams. In our case, we need hour, load, pvpower, pgrid. Here is the photo of how the gms file will look like.

Photo of gms

Now open another gms file and define the sets and parameters of the data and import the data from the gdx file using the command "**$if exist matdata.gms $include matdata.gms**". Gams will import the data from the gdx file and now we can use this data for the optimization.

Optimizing the net price using GAMS:

After importing the data into gams, now we need to define the variables and equations corresponding to the requirements and constraints of the objective. We have taken pcmax, maxstorage, minstorage, chargeff, dischargeff, einitial, one(j) as the parameters.

Pcmax – The maximum rate of charging or discharging of the battery

Maxstorage – Maximum storage capacity of the battery

Minstorage – Minimum storage capacity of the battery

Chargeff – The efficiency with which the battery charges

Dischargeff – The efficiency with which the battery discharges

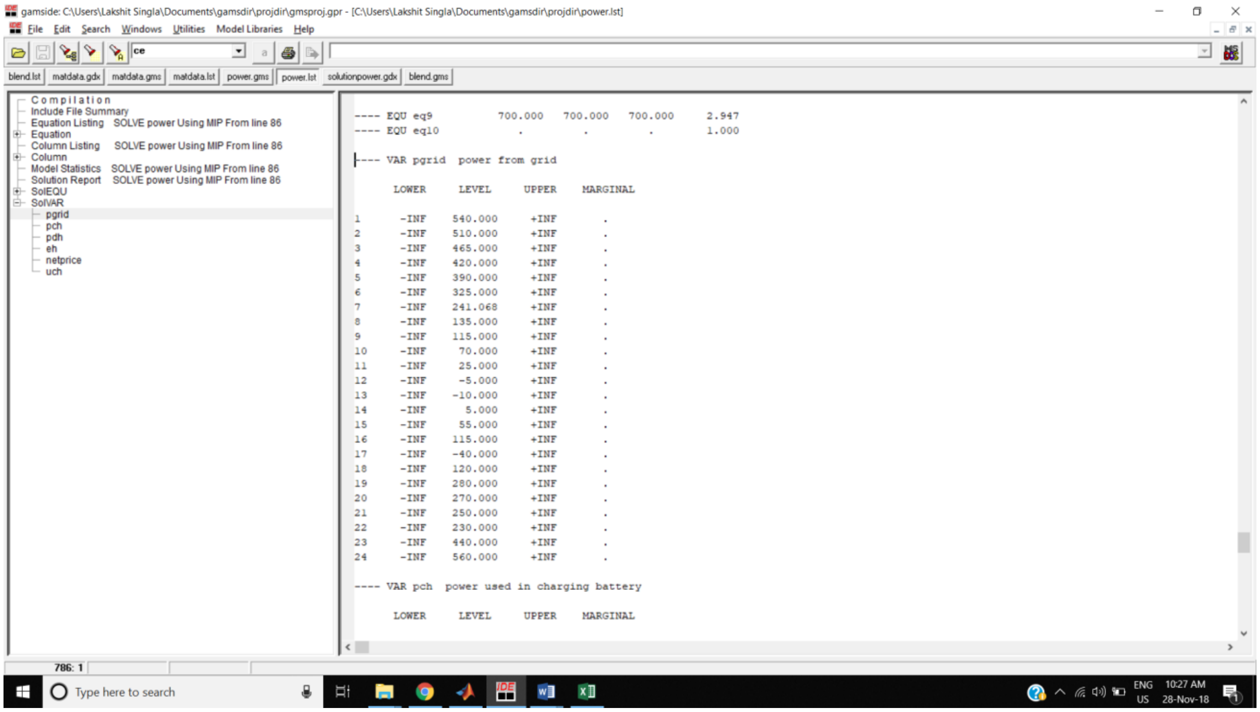
Einitial – The initial battery storage

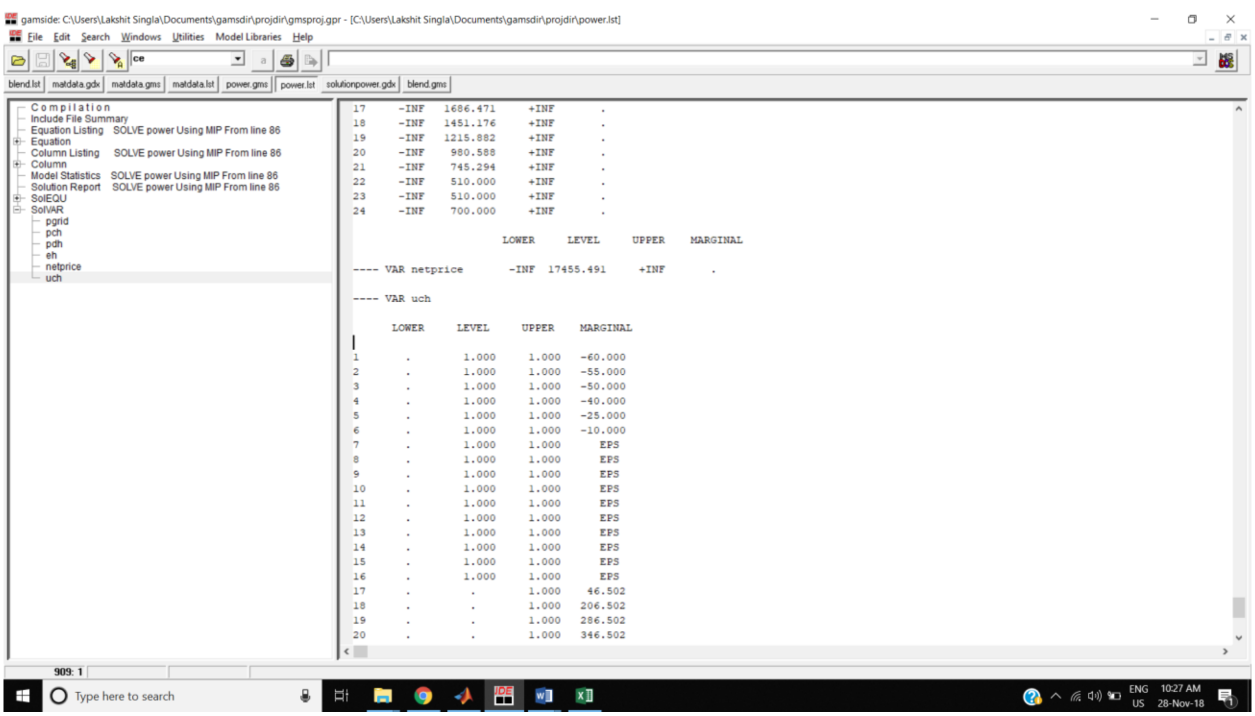
One(j) – to determine whether the battery is being charged or discharged

Then we have defined our variables as pgrid(j), pch(j), pdh(j), eh(i), netprice, uch(j) (binary variable). Here pgrid is the power taken or given to the grid at j^{th} hour, pch and pdh are the charging and discharging efficiencies respectively, eh(i) is the battery power level at every hour. Netprice is the final value paid to the grid and uch is defined to calculate eh at any particular time of the day.

We now write the equations which are used for optimization. The first equation is written for making the total power of the system constant. The second and the third equation describe the limits of the charging power. The fourth and the fifth equation describe the limits of the charging power. Note that here the maximum limit of pch is multiplied with the uch(j) and pdh by (1-uch(j)). This is done because the battery can either charge or discharge at a particular time. Hence uch can be useful to achieve the same thing. Sixth equation gives us the value of power stored in battery at any hour. Equations seven and eight give the minimum and maximum storage of the battery. Equation nine puts a constraint that the initial and final value of the battery must be same. Equation ten calculates the netprice.

Now we optimize the net price using all the above equations and with the help of Mixed Integer Programming. Execute this code. We find the result of our optimization in a new window. It also displays all the variable values. Here is the result of the optimization of the given data.


gamside: C:\Users\Lakshit Singla\Documents\gamsdir\projdir\gmsproj.gpr - [C:\Users\Lakshit Singla\Documents\gamsdir\projdir\power.lst]
File Edit Search Windows Utilities Model Libraries Help
blend.lst matdata.gdx matdata.gms matdata.lst power.gms power.lst solutionpower.gdx blend.gms
Compilation
Include File Summary
Equation Listing SOLVE power Using MIP From line 86
Equation
Column Listing SOLVE power Using MIP From line 86
Column
Model Statistics SOLVE power Using MIP From line 86
Solution Report SOLVE power Using MIP From line 86
SolEQU
SolVAR
pgrid
pch
pdh
eh
netprice
uch
---- EQU eq9 700.000 700.000 700.000 2.947
---- EQU eq10 . . . 1.000
---- VAR pgrid power from grid
LOWER LEVEL UPPER MARGINAL
1 -INF 540.000 +INF .
2 -INF 510.000 +INF .
3 -INF 465.000 +INF .
4 -INF 420.000 +INF .
5 -INF 390.000 +INF .
6 -INF 325.000 +INF .
7 -INF 241.068 +INF .
8 -INF 135.000 +INF .
9 -INF 115.000 +INF .
10 -INF 70.000 +INF .
11 -INF 25.000 +INF .
12 -INF -5.000 +INF .
13 -INF -10.000 +INF .
14 -INF 5.000 +INF .
15 -INF 55.000 +INF .
16 -INF 115.000 +INF .
17 -INF -40.000 +INF .
18 -INF 120.000 +INF .
19 -INF 280.000 +INF .
20 -INF 270.000 +INF .
21 -INF 250.000 +INF .
22 -INF 230.000 +INF .
23 -INF 440.000 +INF .
24 -INF 560.000 +INF .
---- VAR pch power used in charging battery
LOWER LEVEL UPPER MARGINAL
786: 1
Type here to search
ENG US 10:27 AM 28-Nov-18


gamside: C:\Users\Lakshit Singla\Documents\gamsdir\projdir\gmsproj.gpr - [C:\Users\Lakshit Singla\Documents\gamsdir\projdir\power.lst]
File Edit Search Windows Utilities Model Libraries Help
blend.lst matdata.gdx matdata.gms matdata.lst power.gms power.lst solutionpower.gdx blend.gms
Compilation
Include File Summary
Equation Listing SOLVE power Using MIP From line 86
Equation
Column Listing SOLVE power Using MIP From line 86
Column
Model Statistics SOLVE power Using MIP From line 86
Solution Report SOLVE power Using MIP From line 86
SolEQU
SolVAR
pgrid
pch
pdh
eh
netprice
uch
17 -INF 1686.471 +INF .
18 -INF 1451.176 +INF .
19 -INF 1215.882 +INF .
20 -INF 980.588 +INF .
21 -INF 745.294 +INF .
22 -INF 510.000 +INF .
23 -INF 510.000 +INF .
24 -INF 700.000 +INF .
LOWER LEVEL UPPER MARGINAL
---- VAR netprice -INF 17455.491 +INF .
---- VAR uch
LOWER LEVEL UPPER MARGINAL
1 . 1.000 1.000 -60.000
2 . 1.000 1.000 -55.000
3 . 1.000 1.000 -50.000
4 . 1.000 1.000 -40.000
5 . 1.000 1.000 -25.000
6 . 1.000 1.000 -10.000
7 . 1.000 1.000 EPS
8 . 1.000 1.000 EPS
9 . 1.000 1.000 EPS
10 . 1.000 1.000 EPS
11 . 1.000 1.000 EPS
12 . 1.000 1.000 EPS
13 . 1.000 1.000 EPS
14 . 1.000 1.000 EPS
15 . 1.000 1.000 EPS
16 . 1.000 1.000 EPS
17 . . 1.000 46.502
18 . . 1.000 206.502
19 . . 1.000 286.502
20 . . 1.000 346.502
909: 1
Type here to search
ENG US 10:27 AM 28-Nov-18

Here we get the net price to be paid to the grid as Rs 17455.491

Displaying the result in Excel:

We use “$set matout” command to send the optimized values to a gdx file. Later, we can read this gdx file with matlab and using xlswrite, we can write these values into an excel file.