Economic Dispatch Model
For an Electrical Grid with Energy Storage Capability

Eric Anderson

ISyE 720
Professor Jim Luedtke

12/18/2010

Using mixed integer programing techniques, form and solve a mathematical model of economic dispatch on an electrical grid capable of energy storage with a diverse set of power generation assets.
# Table of Contents

Objective ................................................................................................................................. 2

Motivation ............................................................................................................................... 2

Formulation of Energy Storage Problem .................................................................................... 4
  - Critical Ideas to Capture

Mathematical Model of Energy Storage Problem ........................................................................ 5

Forming Data for Problem Instance .......................................................................................... 6
  - Electrical Demand
  - Grid Topology
  - Power Generators

Lagrangian Relaxation ............................................................................................................ 8

Heuristic .................................................................................................................................. 9
  - General Idea
  - Algorithm

Results ...................................................................................................................................... 10
  - Comparing Scenarios
  - Performance

Future Work .............................................................................................................................. 11

Conclusion .............................................................................................................................. 11

Appendix ................................................................................................................................ 12
  - References
  - Sample Model Output
  - Xpress Model Code
  - Relaxation Attempt
Objective

Using mixed integer programming techniques, form and solve a mathematical model of economic dispatch on an electrical grid capable of energy storage with a diverse set of power generation assets.

Motivation

The current structure of the electrical grid requires an exact balance of supply and demand, while being held to a very high standard of reliability. The Independent System Operator is in charge of maintaining the reliability of the grid while also minimizing the cost of power generation by choosing which generators to dispatch.\textsuperscript{[4]} Introducing energy storage to the electrical grid gives the ISO an extra degree of flexibility in operating the grid. This can lead to increased reliability of the grid and lower total cost of power generation.

This model looks at a multiple day time frame, in which the electrical demand is met at all time periods with the lowest possible cost to the system. It is focused on the possible economic gains of having the capability to store energy on the electrical grid. Storing energy can help deal with instantaneous changes in electrical demand, without the need to dispatch expensive generators in order to meet demand. It also has potential to help with grid congestion, which can lead to high variability in spot-prices on wholesale electrical markets. Due to grid congestion, it is possible for wholesale markets to have negative prices at certain buses on the grid. This means that there is a possibility that you would get paid to store energy, and then also get paid when you dispatched it. However, it is not necessary for the markets to have negative prices in order to operate energy storage for a profit. With the high variability of spot prices, it would be possible to take advantage of any significant price differentials at a given bus over time. Lastly, as intermittent power generation (renewables such as wind power) become more prevalent on the electrical grid, it may become necessary to store energy in order to operate these power generation assets in an efficient manner.

Since there are many reasons why energy storage is becoming important to the electrical grid, one must look at the options available to store energy. Certain power generation assets have a natural form to store energy for later use. The simplest version is a hydro-dam with the capability of pumped storage. When the energy markets are in need of energy, the hydro-dam lets the turbines spin to produce energy to sell to the grid. And when the energy markets have too much energy, the hydro-dam can pump water back into the reservoir for later use. This type of energy storage is able to achieve around 70% to 85% round trip efficiency.\textsuperscript{[5]} Another generation asset that has a natural form for energy storage is concentrating solar thermal energy. Using mirrors to concentrate solar irradiation, it is possible to heat a fluid for the eventual end-use of generating steam to spin turbines. This energy can be stored in molten salt reservoirs for use on demand.\textsuperscript{[6]} The last practical energy storage is batteries. With the introduction of plug-in hybrids and electrical vehicles to the car market, the electrical grid is going to have increased electrical demand, but also the possibility to leverage the energy storage of the vehicles for the electrical grid. While the cars have just recently been introduced, there is relatively no energy storage currently available. However, as these cars become main-stream and are deployed in large numbers, they could play a significant role in the capabilities for the electrical grid to store energy.
Lithium-ion batteries can reach round-trip efficiencies of 92%.\textsuperscript{[2]} This comes with a steep upfront cost. However, as these are purchased as a package with the electrical vehicle, any money that could be made from using the batteries as energy storage for the grid would be a net positive for anyone who owns an electrical vehicle. This comes with plenty of logistical problems, but the positives may outweigh the negatives.

With today’s technology, there are multiple ways of efficiently storing energy for use on the electrical grid. Due to how the grid operates, there are many avenues that using energy storage could benefit both the consumer of power through lower energy prices and those operating energy storage for economic gain. Given that there are multiple forms this energy storage may take, no assumptions will be made on the financial cost of energy storage, only that it comes at a cost to efficiency. This means that even though it will be cheaper to meet demand, total energy production will have to increase.

As there are many ways that energy storage can be implemented on the electrical grid, it is important to start looking at the effects this may have. What follows will be a formulation of a minimum cost dispatch model of an electrical grid. First, the main ideas of the formulation will be recognized, and then the mathematical formulation will be shown. Then, data must be formed for the problem instance. All of the data used is randomly generated, with an attempt to capture realities of the electrical grid. The random data will help ensure that the methodologies of studying this problem are robust to separate data instances. Once the data has been formed, an attempt will be made to solve the model optimally. As size of this problem becomes unwieldy as the number of nodes and time periods increase, different methods will be employed to try to speed up the solve time. The results of the model will then be explained, as will the success of the various methods employed. The conclusion will then sum up the significant points of this report.
Formulation of Energy Storage Problem

The basis for this model is the “DC” power flow model. It is an approximation to the true non-linear power flow on an electrical grid. It is reliable for active power flow analysis as long as the following assumptions are met.\(^3\)

- Voltage angle difference are small, \(\sin(\delta) \cong \delta\)
- Line resistance can be neglected, i.e. a lossless line
- Flat voltage profile, avoid deviation from predefined value

These assumptions are decent for a real power grid, and results in an error of under 5%.\(^3\) The following equations form the “DC” power flow model. \(P_i\) is the active power injected into the grid at node i. \(B_{ij}\) is the line susceptance for power flow on a given line. \(\delta_i\) is the voltage angle at node i. \(I\) is the set of all nodes.

\[
P_i = \sum_j B_{ij}(\delta_i - \delta_j) \\
P_{i,\text{gen}} - P_{i,\text{dem}} - P_i = 0
\]

For energy storage to be viable, there must be significant difference in the marginal price of energy over time. The two main reasons for the difference in marginal price are grid congestion and the limited capabilities of generation assets. This study focuses primarily on the effect of having differences in the capabilities of generation assets. The generators will be modeled with both a fixed cost for operating and a per-unit cost for each unit of electricity generated. The other critical factor to model is the limited range of dispatch of a generator over time. This is to represent the fact that a coal plant cannot ramp up production from no energy produced to full capacity in an arbitrarily small amount of time. The first constraint enforces an upper capacity as well as the binary variable \(Y\), which represents whether the generator is on or off. The second constraint represents the range of dispatch over time.

\[
P_{git} \leq Y_{git} U_g \\
P_{git} - \alpha_g \leq P_{git(t+1)} \leq P_{git} + \alpha_g
\]

The range of dispatch constraints give a generator an effective start-up and turn-off time to ramp up and down from full capacity. This time is represented by the following.

\[
t_{\text{eff}} = \frac{U_g}{\alpha_g}
\]

Lastly, to model the capability of energy storage, non-negative variables \(D\) and \(S\) represent the amount of stored energy dispatched and the amount of energy stored, respectively. Then, the following equation links the energy storage over time. \(\eta\) is the round trip efficiency of energy storage.

\[
E_{i(t+1)} = E_{it} - D_{it} + \eta S_{it}
\]

Finally, the “DC” power flow is adapted to include a pseudo-load of energy stored and a pseudo-supply of stored energy dispatched. The complete mathematical model is shown on the following page.
Mathematical Model of Energy Storage Problem

Objective

\[ \text{Minimize } \sum_{t \in T} \sum_{i \in I} \sum_{g \in G} [F_g Y_{git} + V_g P_{git}] \]

Power Flow

1. \[ P_{it} = \sum_{j \in I} B_{ij} (\delta_{it} - \delta_{jt}) - D_{it} \quad \text{for } \forall i \in I, \forall t \in T \]
2. \[ \sum_{g \in G} P_{git} - S_{it} - L_{it} - P_{it} = 0 \quad \text{for } \forall i \in I, \forall t \in T \]

Generators

3. \[ P_{git} \leq Y_{git} U_{git} \quad \text{for } \forall i \in I, \forall t \in T, \forall g \in G \]
4. \[ P_{git} - \alpha_g \leq P_{git(t+1)} \leq P_{git} + \alpha_g \quad \text{for } \forall i \in I, \forall t \in T, \forall g \in G \]
5. \[ P_{git} \geq 0 \quad \text{for } \forall i \in I, \forall t \in T, \forall g \in G \]
6. \[ Y_{git} \in \{0, 1\} \quad \text{for } \forall i \in I, \forall t \in T, \forall g \in G \]

Energy Storage

7. \[ E_{i(t+1)} = E_{it} - D_{it} + \eta S_{it} \quad \text{for } \forall i \in I, \forall t \in T \]
8. \[ E_{it} \geq 0, D_{it} \geq 0, S_{it} \geq 0 \quad \text{for } \forall i \in I, \forall t \in T \]

Sets

\[ I = \{1, 2, \ldots i\} \quad \text{Set of Nodes} \]
\[ G = \{1, 2, \ldots g\} \quad \text{Set of Generators} \]
\[ T = \{1, 2, \ldots t\} \quad \text{Set of Discrete Time Intervals} \]

Variables

\[ P_{it} : \text{Active power leaving node } i \text{ at time } t \]
\[ P_{git} : \text{Active power injected at node } i, \text{ by generator } g, \text{ at time } t \]
\[ Y_{git} : \text{Generator } g \text{ on at node } i \text{ at time } t \text{ if } 1, \text{ else } 0 \]
\[ \delta_{it} : \text{Phase angle relating current to voltage at node } i \text{ at time } t \]
\[ E_{it} : \text{Energy stored at node } i \text{ at time } t \]
\[ S_{it} : \text{Energy stored from the grid at node } i \text{ at time } t \]
\[ D_{it} : \text{Energy dispatched from storage at node } i \text{ at time } t \]

Data

\[ L_{it} : \text{Active power withdrawn from node } i \text{ at time } t \]
\[ B_{ij} : \text{Admittance of grid line from node } i \text{ to node } j \]
\[ F_g : \text{Fixed cost for generator } g \text{ while operating} \]
\[ V_g : \text{Variable cost for generator } g \text{ per unit of energy produced} \]
\[ U_g : \text{Maximum power production from generator } g \]
\[ \alpha_g : \text{Amount of change possible in power production from generator } g \]
\[ \eta : \text{Efficiency of energy storage} \]
Forming Problem Instance Data

Two major factors determining the difficulty of the problem instance are the number of nodes in the grid and the number of time periods. To make the problem of reasonable size, 14 nodes and 72 time periods are used. This means there are approximately \( N \times T \) (1008) binary variables and \( 6 \times N \times T \) (6048) continuous variables. There are three main components to the rest of the data for the problem. They are electrical demand, grid topology, and the power generation assets. All of the data is randomly generated in a way to represent what a real world example would entail. First, the electrical demand will be looked at. Following is the grid topology and finally the power generators.

Electrical demand varies both daily and seasonally. Since this is only a multi-day model, just the daily variation needs to be accounted for. This is done for each node by the following.

- Randomly generate a base-load consumption number, that is the minimum demand over a full 24 hour period.
- Perturb a sine wave function to represent how the demand peaks. Peak power is the maximum demand over the 24 hour period. The sine wave has a 24 hour period, and is then scaled according to a randomly generated peak demand number.
- For each time period, a coin is flipped to see if there is an extra instantaneous demand. If so, a random number is scaled to represent a one-time demand to the system.
- Finally, the base-load, peak demand function, as well as the random single period demands are added together. The following is a sample of a one day demand scenario.

![Energy Demand (1 day sample instance)](image URL)
The real world electrical grid is a graph with sparse edges connecting the nodes. For a given interconnected grid, there are normally no islands. That is, the set of connected components of the graph is all the nodes. For each node, the following is done to form edges.

- Flip coin for every other node and connect the edge with a given probability. This edge is given a random electrical susceptance number, with an upper and lower bound. This number is largely responsible for how electricity is allowed to flow on the network. An analog would be the diameter of a pipe for a gas/fluid pipeline.

- If no edges were connected, a random node is picked to connect with an edge. This is then given a random electrical susceptance number.

Since this does not guarantee the lack of islands, an attempt was made to visualize the grid. In practice, an island was rarely seen. Here is an example of one of the grid topologies used. Since this is a lossless network, there is no significance to the nodes in relationship to some relative geography. The output of each model graphs the grid topology as well as showing the total number of edges in the graph. While it was a goal to create a relatively sparse network, the number of lines as well as the electrical susceptance of each line is critical in determining whether the problem is feasible. The probabilities used to generate the grid were a compromise between creating a sparse grid and one that is usually feasible.

The last set of data is that for the power generators. This data was hand-coded in order to represent a scenario in which energy storage is useful. This was done by creating a set of cheaper generation assets that have low range of dispatch as well as a set of more expensive generation assets that have a high range of dispatch. Generators can be classified into two categories, base load power plants and peaking power plants. The set of generator data chosen was an attempt to capture the characteristics of the real world generators of a coal plant (relatively cheap per-unit cost with low range of dispatched, base load power plant) compared to a natural gas plant (relatively expensive with high range of dispatch, peaking power plant). Lastly, each node is randomly assigned one of the six types of generators, or no generator.

<table>
<thead>
<tr>
<th>Generator type: 1</th>
<th>Generator type: 3</th>
<th>Generator type: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Cost: 30, Variable Cost: 3</td>
<td>Fixed Cost: 60, Variable Cost: 2</td>
<td>Fixed Cost: 100, Variable Cost: 1</td>
</tr>
<tr>
<td>Change in production per time: 10</td>
<td>Change in production per time: 8</td>
<td>Change in production per time: 37</td>
</tr>
<tr>
<td>Max production per generator: 125</td>
<td>Max production per generator: 300</td>
<td>Max production per generator: 600</td>
</tr>
<tr>
<td>Effective start-up time: 12.5</td>
<td>Effective start-up time: 37.5</td>
<td>Effective start-up time: 120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generator type: 2</th>
<th>Generator type: 4</th>
<th>Generator type: 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Cost: 50, Variable Cost: 5</td>
<td>Fixed Cost: 90, Variable Cost: 3.5</td>
<td>Fixed Cost: 150, Variable Cost: 2</td>
</tr>
<tr>
<td>Change in production per time: 30</td>
<td>Change in production per time: 37</td>
<td>Change in production per time: 45</td>
</tr>
<tr>
<td>Max production per generator: 75</td>
<td>Max production per generator: 200</td>
<td>Max production per generator: 400</td>
</tr>
<tr>
<td>Effective start-up time: 2.5</td>
<td>Effective start-up time: 5.4</td>
<td>Effective start-up time: 8.9</td>
</tr>
</tbody>
</table>
Lagrangian Relaxation

The above formulation has several hard constraints. In order to speed up the solve as well as possibly finding a good lower bound, Lagrangian relaxation was performed on two of the constraints separately. The first constraint that could be seen as the hardest is the strict equality of energy balance at each node.

$$\sum_{g \in G} P_{git} - S_{it} - L_{it} = 0$$

Since it is an equality constraint, a free weight $\lambda$ is assigned to penalize not meeting this constraint exactly. The objective function then becomes the following.

$$\text{Minimize } \sum_{t \in T} \sum_{l \in L} \sum_{g \in G} [F_g V_{git} + V_g P_{git}] + \lambda \ast \sum_{l} \sum_{g} (\sum_{g} P_{git} - S_{it} - L_{it} - P_{it})$$

This was a bad idea. Since the grid was capable of storing or dispatching energy, the program either drove energy storage or stored energy dispatch arbitrarily high in order to take advantage of either a positive or negative weight. The program ultimately ended up producing no energy for the nodes to consume.

The other constraint that can be very restrictive is the range of dispatch constraint. This enforces strong conditions on what any given generator is able to do from any given time period to the next.

$$-P_{git(t+1)} + P_{git} - \alpha_g \leq 0$$
$$P_{git(t+1)} - P_{git} - \alpha_g \leq 0$$

Then assigning each constraint a non-negative weight $\lambda$ and $\mu$, the constraints can be moved into the objective. The following objective penalizes violation of the constraints.

$$\text{Minimize } \sum_{t \in T} \sum_{l \in L} \sum_{g \in G} [F_g V_{git} + V_g P_{git} + \lambda(-P_{git(t+1)} + P_{git} - \alpha_g) + \mu (P_{git(t+1)} - P_{git} - \alpha_g)]$$

Equivalently,

$$\text{Minimize } \sum_{t \in T} \sum_{l \in L} \sum_{g \in G} [F_g V_{git} + V_g P_{git} + (\mu - \lambda)P_{git(t+1)} + (\lambda - \mu)P_{git} - (\lambda + \mu)\alpha_g]$$

This didn’t work either. The lower bound was much worse than those found by the solver. The symmetric nature of the constraints led to an awkward penalty function. The objective was systematically lowered by the negatively weighted $\alpha_g$ values.

Next, an attempt was made to relax the range of dispatch constraint while not penalizing the objective function. This attempt led to a much better lower bound (one instance gave a lower bound within 2%). However, the problem still appeared to be computationally difficult. The solver was not able to consistently find a proven optimal solution of the relaxation in significantly less time.

The source code for the main function in the attempt to find a decent relaxation is in the appendix.
Heuristic

In solving the model, it became readily apparent that finding feasible solutions is a hard part of the problem. In the various solves performed, it was seen that only around 5 to 10 feasible solutions were found, and never more than 20. Because of this, a heuristic which develops feasible solutions could be useful. Following will be an outline of a possible algorithm that may develop quality feasible solutions. However, the heuristics has not been implemented, so the quality of the heuristics is unknown.

Since the model is capable of storing energy, an obvious solution would be to generate the same amount of energy each time period, and then redistribute the energy through storage based on the given demand in a time period. After this solution is found, iteratively improve it by finding time periods in which the marginal cost of electricity is lower than the average cost of electricity for the base scenario. Then, increase production at that time period, and lessen the constant production over time by an equivalent amount. This is somewhat complicated due to the efficiency of energy storage not being equal to 1 (we live in the real world).

Step 0:
- Find the constant production needed per time period to satisfy total demand
  - This can be done iteratively, first find average electrical demand
  - For each time $t$, look at difference between demand and average electrical demand
  - Scale the stored energy by the efficiency, and look at difference from dispatched energy
  - Divide this difference by $T$ (total periods) and add this value to average electrical demand
  - Repeat until scaled stored energy is arbitrarily close to dispatched energy
- Find the optimal dispatch for this demand

Step k:
- Look at time periods in which demand is greater than constant production
  - Effective marginal cost is then average cost divided by efficiency, because energy is being dispatched from storage to meet the last unit of demand
  - Then calculate marginal cost from increasing generator production at that time period
  - If actual marginal cost is less than effective marginal cost, increase production at that time period until the two become close
  - Take the energy displaced from increased production and divide by efficiency to find actual energy displaced
  - Divide this figure by $T$, and lower constant production by this amount
- Done once effective marginal cost and actual marginal cost are arbitrarily close for all time periods
Results

The model as formed proved to be hard to solve in a timely manner. The large number of variables and tough constraints made finding feasible solutions difficult. The following table summarizes the results of five separate data runs. Then, an explanation of the results will follow.

<table>
<thead>
<tr>
<th>Solve #</th>
<th>Energy Storage</th>
<th># of Lines</th>
<th># of Gen. 1</th>
<th># of Gen. 2</th>
<th># of Gen. 3</th>
<th># of Gen. 4</th>
<th># of Gen. 5</th>
<th># of Gen. 6</th>
<th>Avg. Unit Cost</th>
<th>Solve Time (s)</th>
<th>Optimality Gap (%)</th>
<th>Nodes Processed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>22</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.32</td>
<td>1000</td>
<td>.0135</td>
<td>106557</td>
</tr>
<tr>
<td>1</td>
<td>No</td>
<td>22</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.60</td>
<td>1000</td>
<td>.0391</td>
<td>139349</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>28</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1.19</td>
<td>1000</td>
<td>.0131</td>
<td>154608</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>28</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1.55</td>
<td>1000</td>
<td>.0226</td>
<td>172197</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>22</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1.85</td>
<td>1000</td>
<td>.0040</td>
<td>64067</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>22</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1.95</td>
<td>1000</td>
<td>.0104</td>
<td>145837</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>27</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.76</td>
<td>20.7</td>
<td>1e-5</td>
<td>264</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>27</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.90</td>
<td>1000</td>
<td>.0057</td>
<td>152631</td>
</tr>
<tr>
<td>5</td>
<td>Yes</td>
<td>27</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1.30</td>
<td>1000</td>
<td>.0133</td>
<td>24002</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>27</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1.41</td>
<td>1000</td>
<td>.0311</td>
<td>129012</td>
</tr>
</tbody>
</table>

As can be seen, almost all of the runs hit the time limit before proving an optimal solution. A typical run started out with large gains in both feasible solutions as well as improved lower bounds for the beginning iterations. As the number of iterations increased, the model begins to stagnant and reduces to an almost complete enumeration of feasible solutions. However, even without proven optimal solutions, a number of conclusions can be drawn from the results of these runs.

First, looking at the computational difficulty of solving the problem with versus without energy storage, the problem is easier with the capability of energy storage. This is not surprising. When the model is able to store energy, the range of dispatch constraints become less important. Since the grid can store energy, it is not necessary for the model to meet demand exactly at every time period. Typically, the model finds an optimal power output of the generators and tries to operate at that level for the majority of the time. Then, at a cost to efficiency, the model redistributes energy from the low demand time periods to the high demand time periods.

Next, the difference in average unit cost of electricity is important to note. While the generator data is not from the real world and may be exaggerated to some extent, the model was able to consistently find improvements in total cost of dispatch. While this does not prove that energy storage will be economically feasible in the real world, it is strong evidence to suggest that the situation should be investigated further.

Lastly, a note should be made on solve #4, which was able to find an optimal solution. The total energy demand was less than usual for this problem. As such, the model was able to operate a cheap generator constantly, and redistribute the energy without worries of a cheaper solution.
Future Work

The problem this study focused on is one in which there is no uncertainty. The demand is known for all time periods, as well as the availability of generators and storage capabilities. An interesting extension of this would be to look at the problem from ISO’s perspective. That is, current demand is known and there is a probability distribution of future demand. Then, the ISO’s problem would be to minimize the cost of current generation dispatch as well as the expected value of future generation dispatch. This would be a stochastic programming model. It would help shed light on how the ISO is supposed to operate when they have the capability to store and/or dispatch stored energy now, while also minimizing the expected value of the future cost of power generation.

Also, other work could be done on this model to help quantify the real economic benefits of energy storage. While this model shows that energy storage could play a large role in decreasing total cost of electrical generation, no attempt was done to use real world data. A real world grid topology as well as a set of generating assets for that given grid would be necessary to put real world numbers into this framework. Energy storage would also need to be modeled more precisely. That is, there may be some fixed cost for constructing the storage capabilities, as well as possible operating and maintenance cost. Then, the model would more accurately show the potential economic benefits of being able to store energy on an electrical grid. To counter the complexity of the problem, a single day model could be used along with the constraints that the energy stored at the beginning of the day is equal to that stored at the end of the day.

Conclusion

The model formed for this study was able to effectively show that this situation is worth further study. While it most certainly has its shortcomings, it is able to show the major factors that play into the dynamics of energy storage on the electrical grid. The main factor in determining the effectiveness for energy storage is a difference in marginal prices over time. This study looked at how the range of dispatch of generators can lead to these price differences. Since the figures used for the generator data were hypothetical, no conclusion can be made to the level of actual economic gains possible of using energy storage. However, it is probable that due to the pervasiveness of electricity in our world, there are many situations in which energy storage could be operated for economic benefits.

The major downside to this model is complexity. Even with a small grid size of 14 nodes, it took too long to solve the problem optimally. While the relaxation approaches attempted were not helpful, the heuristic suggested could prove useful to finding feasible solutions. Also, in the future work section a note was made that the study need not be multiple day, only single day. The daily variation in electrical demand is what energy storage is capable of smoothing. So, with a simple constraint that the amount of energy stored at the beginning of the day is equal to that of the end of the day, only 24 time periods are needed, versus the 72 used in this model. Also, the model used random data, which could cause the problems to be more difficult than a real world instance might be. Lastly, the model is not made to be used as an operation tool for operators of the grid, but to show the economics of energy storage on the grid. As such, the time to solve the model is of less importance.
References

[1] Electrical Demand

http://www.area-alliance.org/documents/base%20load%20power.pdf


http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6TH1-43X7V64-7&_user=443835&_coverDate=10%2F15%2F2001&_rdoc=1&_fmt=high&_orig=search&_origin=search&_sort=d&_docanchor=&view=c&_searchStrId=1583786601&_rerunOrigin=google&_acct=C000020958&_version=1&_urlVersion=0&_userid=443835&md5=a04e291c436a9743bf27c7ea&searchtype=a

[3] DC Power Flow


http://pjm.com/home.aspx

[5] Pumped Hydro

http://www.electricitystorage.org/site/technologies/pumped_hydro/


Output of Model (one instance)

<table>
<thead>
<tr>
<th>No Energy Storage Allowed</th>
<th>Energy Storage Capacity: 1750</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes: 14</td>
<td>Number of nodes: 14</td>
</tr>
<tr>
<td>Number of lines: 21</td>
<td>Number of lines: 21</td>
</tr>
<tr>
<td>Duration: 72 hours</td>
<td>Duration: 72 hours</td>
</tr>
<tr>
<td></td>
<td>Total Energy Consumed: 98220.2</td>
</tr>
<tr>
<td></td>
<td>Total Energy Produced: 98220.2</td>
</tr>
<tr>
<td></td>
<td>Total Cost of Energy Produced: 164832</td>
</tr>
<tr>
<td><strong>Per Unit Cost of Energy:</strong> 1.67818</td>
<td><strong>Per Unit Cost of Energy:</strong> 1.21577</td>
</tr>
</tbody>
</table>

Generator type: 1
Fixed Cost: 30, Variable Cost: 3
Change in production per time: 10
Max production per generator: 125
Number Deployed: 2
Energy Produced: 2717.65
Avg. Production per unit time: 37.7451
Variability in generation: 4.67676

Generator type: 2
Fixed Cost: 50, Variable Cost: 5
Change in production per time: 30
Max production per generator: 75
Number Deployed: 3
Energy Produced: 1402.12
Avg. Production per unit time: 19.4739
Variability in generation: 6.07023

Generator type: 3
Fixed Cost: 60, Variable Cost: 2
Change in production per time: 8
Max production per generator: 300
Number Deployed: 2
Energy Produced: 4166.09
Avg. Production per unit time: 57.8624
Variability in generation: 6.41265

Generator type: 4
Fixed Cost: 90, Variable Cost: 3.5
Change in production per time: 37
Max production per generator: 200
Number Deployed: 0
Energy Produced: 0
Avg. Production per unit time: 0
Variability in generation: 0

Generator type: 5
Fixed Cost: 100, Variable Cost: 1
Change in production per time: 5
Max production per generator: 600
Number Deployed: 3
Energy Produced: 76032.8

Generator type: 1
Fixed Cost: 30, Variable Cost: 3
Change in production per time: 10
Max production per generator: 125
Number Deployed: 2
Energy Produced: 0
Avg. Production per unit time: 0
Variability in generation: 0

Generator type: 2
Fixed Cost: 50, Variable Cost: 5
Change in production per time: 30
Max production per generator: 75
Number Deployed: 3
Energy Produced: 0
Avg. Production per unit time: 0
Variability in generation: 0

Generator type: 3
Fixed Cost: 60, Variable Cost: 2
Change in production per time: 8
Max production per generator: 300
Number Deployed: 2
Energy Produced: 0
Avg. Production per unit time: 0
Variability in generation: 0

Generator type: 4
Fixed Cost: 90, Variable Cost: 3.5
Change in production per time: 37
Max production per generator: 200
Number Deployed: 0
Energy Produced: 0
Avg. Production per unit time: 0
Variability in generation: 0

Generator type: 5
Fixed Cost: 100, Variable Cost: 1
Change in production per time: 5
Max production per generator: 600
Number Deployed: 3
Energy Produced: 100106
Avg. Production per unit time: 1056.01
Variability in generation: 1.35824

Generator type: 6
Fixed Cost: 150, Variable Cost: 2
Change in production per time: 45
Max production per generator: 400
Number Deployed: 1
Energy Produced: 13901.5
Avg. Production per unit time: 193.076
Variability in generation: 9.44763

Avg. Production per unit time: 1390.37
Variability in generation: 1.43515

Generator type: 6
Fixed Cost: 150, Variable Cost: 2
Change in production per time: 45
Max production per generator: 400
Number Deployed: 1
Energy Produced: 0
Avg. Production per unit time: 0
Variability in generation: 0

Performance of Solve
Stop Status: Time Limit Hit
Time: 500.483
Nodes Processed: 90418
Objective Value: 164832
Best Bound: 160689
Relative Optimality Gap: -0.0251332

Performance of Solve
Stop Status: Solution is Optimal
Time: 12.364
Nodes Processed: 409
Objective Value: 121706
Best Bound: 121706
Relative Optimality Gap: 0
model EnergyStorage
uses "mmxprs", "mmive", "mmsystem"

forward procedure store_sol
forward procedure print_sol
forward procedure graph_grid
forward procedure init_solver

declarations
N = 14  ! Number of nodes
T = 72  ! Time
G = 6  ! Generators
rN = 1..N  ! Set (range) of nodes
rT = 1..T  ! Set (range) of time (in half-hours)
rG = 1..G  ! Set (range) of generators
Fixed: array(rG) of real  ! Cost to run given generator
Vari: array(rG) of real  ! Variable cost per power generating
U: array(rG) of real  ! Maximum capacity of generator
alpha: array(rG) of real  ! Ability to change production over time
numberGen: array(rG) of real  ! Number of generators deployed
Load: array(rN, rT) of real  ! Load on given node
B: array(rN, rN) of real  ! Admittance of edge in power network
P: array(rN, rT) of mpvar  ! Power injected into the grid at node n during time t
del: array(rN, rT) of mpvar  ! Phase angle at given node (determines power flow, for "DC"
Pgen: dynamic array(rN, rG, rT) of mpvar  ! Total power generator in given generator g at node n during time t
Y: dynamic array(rN, rG, rT) of mpvar  ! Total storage capacity
S: dynamic array(rN, rT) of mpvar  ! Energy dispatched from storage at node n at time t
Y: dynamic array(rN, rG, rT) of mpvar  ! Energy stored at node n at time t
Z: dynamic array(rN, rT) of mpvar  ! Energy stored from power grid
cap: dynamic array(rN) of real  ! Storage capacity
capS: real  ! Total storage capacity
D: dynamic array(rN, rT) of mpvar  ! Energy dispatched from storage at node n at time t
S: dynamic array(rN, rT) of mpvar  ! Energy stored from power grid
end-declarations

!PROBLEM DATA-------------------------------
eta = .90  ! Efficiency of energy storage

!Generators
Fixed:: [30, 50, 60, 90, 100, 150]
Vari:: [3, 5, 2, 3.5, 1, 2]
U:: [125, 75, 300, 200, 600, 400]
alpha:: [10, 30, 8, 37, 5, 45]
forall(g in rG) numberGen(g):=0
forall(n in rN) do
  gen:=random
  if(gen<=.16) then gen:=1
  elif(gen<=.34) then gen:=2
  elif(gen<=.46) then gen:=3
  elif(gen<=.56) then gen:=4
  elif(gen<=.68) then gen:=5
  elif(gen<=.78) then gen:=6
  else gen:=0
  end-if
  if(gen<>0) then
    numberGen(round(gen)):=numberGen(round(gen)) + 1
    forall(t in rT) do
      create(Y(n,round(gen),t))
      create(Pgen(n,round(gen),t))
    end-do
  end-if
end-do

!Energy storage
forall(n in rN) do
  if(random<.5) then
    forall(t in rT) do
      create(E(n,t))
      create(S(n,t))
      create(D(n,t))
    cap(n) := 250
  end-do
end-if
end-do

capS:=sum(n in rN) cap(n)

!Load
baseload:=70
forall(n in rN) do
baseN:=baseload*random
peakN:=peak*random
peakT:=(random-.5)*1.2 + M_PI
forall(t in rT) do
  rand1:=round(random)*random*65
  rand2:=round(random)*random*55
  rand3:=round(random)*random*35
  Load(n,t):=baseN + peakN*(cos(t*M_PI/12 - peakT)+1) + rand1 + rand2 + rand3
end-do
Load(n,1):=(Load(n,T) + Load(n,1))/2
forall(t in 2..T) do
  Load(n,t):= (Load(n,t-1) + Load(n,t))/2
end-do
end-do

!Transmission Grid
numLines:=0
forall (i in rN) do
  line:=0
  forall (j in i..N) do
    bb:= random
    if(bb>=.5 and random<=.40) then
      !First inequality makes sure no extremely small lines and sec
      B(i,j) := 225*bb
      B(j,i) := 225*bb
      line:=1
      numLines:=numLines+1
    end-if
  end-do
  if(line=0) then
    jline:=round(random*N)  !At least one line out of a node
    if(jline = i) then
      jline:=jline+1
      if(jline>N) then
        jline:=1
      end-if
    end-if
    B(i,jline) := 175
    B(jline,i) := 175
    numLines:=numLines+1
  end-if
end-do
!END PROBLEM DATA-----------------------------------

!MODEL-----------------------------------------------
forall (n in rN) E(n,1) = E(n,T)
forall (t in rT) do
forall (n in rN) do
  ! Power Flow Constraints
  P(n,t) = sum(i in rN) ( B(n,i)*(del(n,t) - del(i,t)) ) - D(n,t)
  sum(g in rG) Pgen(n, g, t) - S(n, t) - Load(n, t) = P(n, t)

  del(n, t) is_free  ! to allow power to travel in either direction
  P(n, t) is_free  ! Power into grid can be positive negative
  E(n,t)<cap(n)

! Energy Storage
if(t>T) then
  E(n, t+1) = E(n, t) - D(n,t) + eta*S(n,t)
end-if
forall (g in rG) do
  ! Max Production from generators
  Pgen(n, g, t) <= Y(n, g, t)*U(g)
  if(t<T) then
    Pgen(n,g,t+1) <= Pgen(n,g,t) + alpha(g)
    Pgen(n,g,t+1) >= Pgen(n,g,t) - alpha(g)
  end-if
  Y(n, g, t) is_binary  ! Make on/off switch binary
end-do
end-do
end-do

! Objective function: Minimize cost
CostFunction := \sum_{t \in rT} ( \sum_{n \in rN} ( \sum_{g \in rG} ( \text{Fixed}(g) \cdot Y(n,g,t) + \text{Vari}(g) \cdot Pgen(n,g,t)) )

! Problem declaration finished-------------------------------

! Prepare to run program------------------------------------
declarations
MAXTIME = 500 ! Maximum time
CUTSTRAT = 3 ! Cut strategy to use
CVRCUT = 1 ! Cover cuts on/off
STRBRNCH = 1 ! Strong branching parameter
PSOLVE = 1 ! Presolve parameters

! Performance Stored Results
Time: real ! Time Taken
NodesP: real ! Nodes Processed
ObjV: real ! Objective value
BestB: real ! Best Bound
Ogap: real ! Optimality gap
StopS: real ! Stop Status

! Metrics on electrical grid
loadProfile: array(rT) of real ! Total energy demand from all nodes
avgLoad: real ! Average load per unit time
variLoad: real ! Variability of load
genProfile: array(rT) of real ! Total energy produced from all nodes
avgTGen: real ! Average total gen per unit time
variTGen: real ! Variability of total gen
storedEnergy: real ! Total energy stored
storedProfile: array(rT) of real ! Stored Energy
energyConsumed: real ! Total energy consumed
energyProduced: real ! Total energy produced
energyCost: real ! Cost of energy produced
costPerUnit: real ! Cost per unit of energy produced
produceGen: array(rG) of real ! Production from given generators
avgGen: array(rG) of real ! Average production from given generators
variGen: array(rG) of real ! Variability of production from given gen

end-declarations

! READY FOR SOLVE----------------
forall(n in rN, t in rT) do
fxE(n,t):= E(n,t)=0
fxS(n,t):= S(n,t)=0
fxD(n,t):= D(n,t)=0
end-do

init_solver ! Give solver the parameters
minimize(CostFunction) ! Solve Problem
store_sol ! Store the solution details
forall(n in rN, t in rT) do
sethidden(fxE(n,t), true)
sethidden(fxS(n,t), true)
sethidden(fxD(n,t), true)
end-do

minimize(CostFunction)
store:=1
print_sol ! Output results of trials
store_sol
graph_grid
store:=0
print_sol

!------PROCEDURES------!

! Graph nodes and electrical connection
procedure graph_grid
declarations
graph: integer
a: array(rN) of real
b: array(rG) of real
end-declarations
forall(i in rN) do
a(i):=round(i/4 )
forall(i in rN) do
  IVEdrawpoint(graph, a(i), b(i))
end-do

forall(i in rN, j in rN | B(i,j) >= 1) do
  IVEdrawline(graph, a(i), b(i), a(j), b(j))
end-do

forall(i in rN) do
  procedure
    ! Print out all results
  end-procedure

procedure store_sol
  ObjV := getobjval
  BestB := getparam("XPRS_BESTBOUND")
  StopS := getparam("XPRS_STOPSTATUS")
  NodesP := getparam("XPRS_NODES")
  Time := gettime
  Ogap := (BestB - ObjV)/ObjV
  forall(t in rT) do
    loadProfile(t):= sum(n in rN, Load(n,t))
    genProfile(t):= sum(n in rN, g in rG, getsol(Pgen(n,g,t))
    storedProfile(t):= sum(n in rN, getsol(E(n,t)))
  end-do
  storedEnergy:= sum(n in rN, t in rT) getsol(S(n,t))
  energyConsumed:= sum(n in rN, t in rT) Load(n,t)
  energyProduced:= sum(n in rN, t in rT, g in rG) getsol(Pgen(n,g,t))
  energyCost:= ObjV
  costPerUnit:=energyCost/energyProduced
  avgLoad:= sum(t in rT) loadProfile(t)
  avgTGen:= sum(t in rT) genProfile(t)
  variLoad:=0
  forall(t in rT) do
    variLoad:=variLoad + (genProfile(t) - avgLoad )^2
  end-do
  variLoad:= sqrt(variLoad/energyConsumed)
  variTGen:=0
  forall(t in rT) do
    variTGen:=variTGen + (genProfile(t) - avgTGen )^2
  end-do
  variTGen:= sqrt(variTGen/energyProduced)
  forall(g in rG) do
    produceGen(g):=sum(n in rN, t in rT) getsol(Pgen(n,g,t))
    if(produceGen(g)> .01) then
      avgGen(g):= produceGen(g)/T
      variGen(g):=0
      forall(t in rT) do
        variGen(g):=variGen(g) + (sum(n in rN) getsol(Pgen(n,g,t)) - avgGen(g) )^2
      end-do
      variGen(g):= sqrt(variGen(g)/produceGen(g))
      end-if
  end-do
end-procedure

procedure print_sol
  writeln(" ")
  writeln(" ")
  writeln("Energy Storage on an Electrical Grid")
  writeln(" ")
writeln("Variability in Energy Production: ", variTGen)
writeln("Average Energy Production per Time: ", avgTGen)
writeln("Variability in Electrical Load: ", variLoad)
writeln("Average Electrical Load per Time: ", avgLoad)
writeln("Variability in Energy Production: ", variGen)
if(store=1) then writeln("No Energy Storage Allowed")
else writeln("Energy Storage Capacity: ", capS)
end-if

procedure
  Initialize a trial run by setting solver parameters
setparam("XPRS_MAXTIME", MAXTIME)
setparam("XPRS_verbose", true)
setparam("XPRS_CUTSTRATEGY", CUTSTRAT)
setparam("XPRS_COVERCUTS", CVRCUT)
setparam("XPRS_PRESOLVE", PSOLVE)
setparam("XPRS_SBEFFORT", STRBRNCH)
end-procedure

! Initialize a trial run by setting solver parameters
procedure init_solver
setparam("XPRS_MAXTIME", MAXTIME)
setparam("XPRS_verbose", true)
setparam("XPRS_CUTSTRATEGY", CUTSTRAT)
setparam("XPRS_COVERCUTS", CVRCUT)
setparam("XPRS_PRESOLVE", PSOLVE)
setparam("XPRS_SBEFFORT", STRBRNCH)
end-procedure

end-procedure
Objective Functions

CostFunctionRelaxGen := sum(t in rT) ( sum(n in rN) ( sum(g in rG) ( Fixed(g) * Y(n,g,t) + Vari(g) * Pgen(n,g,t) ))) + sum(t in 1..T-1) ( sum(n in rN) ( sum(g in rG | Yreal(n,g,t) = 1) ( (lambda-mu)*Pgen(n,g,t) + (mu-lambda)*Pgen(n,g,t+1) - (mu+lambda)*alpha(g) )))

CostFunctionRelaxBalance := sum(t in rT) ( sum(n in rN) ( sum(g in rG) ( Fixed(g) * Y(n,g,t) + Vari(g) * Pgen(n,g,t) ))) + lambdaBalance* ( sum(t in rT, n in rN, g in rG) ( Pgen(n,g,t) - S(n,t) - Load(n,t) - P(n,t) ))

! Main Loop for Lagrangian Relaxation Attempt
! READY FOR SOLVE----------------
init_solver
minimize(CostFunction) ! Give solver the parameters
store_sol
print_sol ! Solve Problem
graph_grid ! Store the solution details

! Relax range of dispatch generator constraint
forall(n in rN, t in 1..T-1, g in rG) do
  sethidden(genup(n,g,t), true)
  sethidden(gendown(n,g,t), true)
end-do

! Solve problem with penalized objective
minimize(CostFunctionRelaxGen)
store_sol
print_sol

! Solve problem without penalized objective
minimize(CostFunction)
store_sol
print_sol

! Enforce range of dispatch again
forall(n in rN, t in 1..T-1, g in rG) do
  sethidden(genup(n,g,t), false)
  sethidden(gendown(n,g,t), false)
end-do

! Relax energy balance constraint
forall(n in rN, t in rT) do
  sethidden(balance(n,t), true)
end-do
minimize(CostFunctionRelaxBalance)
store_sol
print_sol