The Unit Commitment Problem

JIM OSTROWSKI

Industrial and Systems Engineering
University of Tennessee jostrows@utk.edu

BEN KNUEVEN

Industrial and Systems Engineering
University of Tennessee

JIANHUI WANG

Argonne National Laboratory
JP WATSON

Sandia National Laboratory

The Basic Problem

The UC Problem

$$\text{Minimize} \sum_{t \in T} \sum_{j \in J} c^j(p^j_t)$$

subject to

$$\sum_{j \in J} p^j_t \geq D_t, \quad \forall\, t \in T$$

$$p^j \in \Pi^j, \quad \forall j \in J.$$

- $c(p^j_t)$ gives the cost of generator j producing p^j_t units of electricity at time t.
- In every time periods, demand D_t must be met.
- Each generator must work within its physical limits (ramping constraints, minimum shut down times, etc.).

Physical Constraints of Generators

- **Convex Production Costs**
- **Minimum & Maximum Output Levels:** If the generator is on, it must produce between $\underline{\mathrm{P}}$ and $\overline{\mathrm{P}}$ units of power.
- **Ramping Constraints:** Power output cannot change too rapidly over a short period of time.
- **Minimum Up (Down) Time:** When a generator is turned on (off), it must stay on for at least UT (DT) time units.
- **Downtime Dependent Startup Costs:** The cost of turning on a generator is dependent on how long the generator has been off.

Basic Approach

- A strategy employed by many researchers is to investigate tight formulations for a generic generator, i.e., tight descriptions of Π.
- This work will employ the same tactic.

Main Result:

We will give a tight and compact (convex hull) description of the feasible operating schedule of a generator. Moreover, this description is fairly flexible and can enable a variety of additional physical constraints

A Brief Outline

- First, we will discuss some previous work on polyhedral results related to electric generator schedules.
- Then, we will move into more general polyhedral description.
- Lastly, we discuss how to model cases when there are similar (and almost similar) generators.

Polyhedral Results for Generator Scheduling

"1-binary variable model"

- I can write the feasible region of a generator using two variables per time period.
- Let p_t be the (continuous) variable representing power output.
- Let u_t be the (binary) variable representing if the generator is on/off.
- The convex hull description of this polyhedron is known *if there is no ramping constraint*, but it is *large* (exponential).
- But, a polynomial-time cutting-plane method exists (Lee, Lueng, Margot: 2004).

3 Binary Variable Model

3-Bin

- Now we use 4 variables per time period:
- Let p_t be the (continuous) variable representing power output.
- Let u_t be the (binary) variable representing if the generator is on at time t.
- Let v_t be the (binary) variable representing if the generator is turned on at time t.
- Let w_t be the (binary) variable representing if the generator is turned off at time t.

- Yes, the additional variable are redundant. But, they allow us to write tight descriptions of the polytope *with no ramping constraints* (Rajan & Takriti: 2005).

Not Quite the Same Thing, but Nice

- A slightly different approach to generator scheduling comes from Frangioni and Gentile, who solve the single unit commitment problem (1UC) in polynomial time using dynamic programming.
  - The 1UC model assumes prices are fixed, then optimizes a single unit's profit.
- The trick: Since the prices are known, it is easy to compute the exact production schedule at times in the interval $[a, b]$ if is is known for sure that the generator turns on at time a and then shuts down at time b (Economic Dispatch Problem).
- There are at most Tc2 many valid turn on/turn off time intervals, so you only need to consider combining the corresponding production schedules, where the only constraint is the minimum downtime constraint.

Economic Dispatch Problem

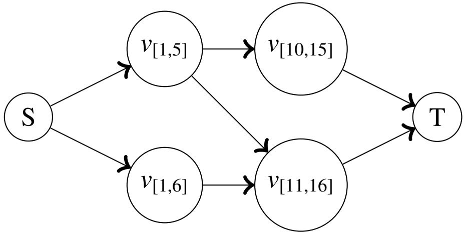
- If it is know that the generator is turned on at a and off at b, the profit during this time period is solved via the *linear program*:

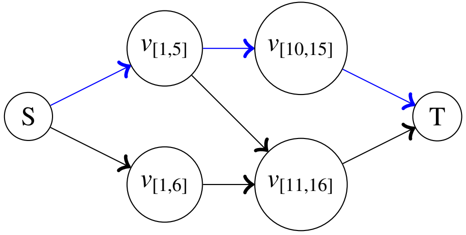
$$
\begin{aligned}
p_i^{[a,b]} &\leq 0 && \forall i < a \text{ and } i > b \\
-p_i^{[a,b]} &\leq -\underline{P} && \forall i \in [a,b] \\
p_i^{[a,b]} &\leq \min(\overline{P}, SU + (i-a)RU, SD + (b-i)RD) && \forall i \in [a,b] \\
p_i^{[a,b]} &\leq p_{i-1}^{[a,b]} \min(RU, SU + (b-i)RD - \underline{P}) && \forall i \in [a+1,b] \\
p_{i-1}^{[a,b]} &\leq p_i^{[a,b]} + \min(RD, SU + (i-a)RU - \underline{P}) && \forall i \in [a+1,b].
\end{aligned}
$$

A Dynamic Programming Approach to 1UC

- The 1UC model is solved via a shortest path problem in the following digraph:
- Let s be the source node, t be the sink node.
- Let $v_{[a,b]}$ represent the action of turning on the generator at time a and shutting it off at time b. The cost of going through node $v_{[a,b]}$ is equal to negative the profit from the economic dispatch problem.
- There is an arc leaving (entering) s (t)and entering (leaving) ever other vertex.
- Arc $(v_{[a,b]}, v_{[c,d]})$ exists if $b + \texttt{mindowntime} \leq c$.
- Digraph is acyclic, shortest path is easily found.

An Example: Min Up/Downtime=5





Remarks:

- The dynamic programming approach to 1UC is a fantastic result, but it hasn't been very helpful for multi-generator models.
- Why? The DP only considers Tc2 specific schedules, not all possible production schedules. There hasn't been an obvious way of extending this idea to more general methods.

Fundamental Problem:

Extend this result to general UC models.

How This Helps UC

- The dynamic programming problem, provides framework that allows us to build a schedule by visiting different nodes in the graph.
- If I visit node $v_{[a,b]}$, I can produce in periods $[a, b]$.
- However, I have constraints on how I build my solution! If I visit $v_{[a,b]}$ I cannot visit $v_{[a+1,b]}$!
- This restriction can be modeled by adding constraints on the γ terms (where γ represents if I visit a node or not).

Sums of Dispatch Polytope

- Let $\gamma_{[a,b]}$ represent if the generator is on during the interval $[a, b]$.

Generator Polytope

$$D \stackrel{\text{def}}{=} \begin{cases} A^{[a,b]} p^{[a,b]} \leq b^{[a.b]} \gamma^{[a,b]} & \forall [a,b] \in \mathcal{T} \\ \displaystyle\sum_{[a,b] \in \mathcal{T}} p^{[a,b]} = p & \\ \displaystyle\sum_{\{[a,b] \in \mathcal{T} \mid i \in [a, b + \mathtt{mindowntime}]\}} \gamma^{[a,b]} \leq 1 \ \forall i. \in T & \\ \gamma^{[a,b]} \geq 0 & \\ p^{[a,b]} \in \mathbb{R}^n_+. & \end{cases}$$

Remarks

- There is a compact & tight formulation for generators. Moreover, this a very general framework. Any additional constraints can be added so long as Γ remains integer and the feasible dispatch problem remains a polytope.
- Allows for:
  - Arbitrary startup costs
  - On-time dependent ramping constraints (to model startup and shutdown trajectories)
  - Multistage Stochastic UC
  - and more!
- Cons of this approach:
  - Tight but large! Tc3 many variables per generator. (Though only T many binomial variables are required).

Lift and Project Cuts

- Using the full model results in a huge linear programming problem. The LP takes too long to solve!
- Another idea is to use the 3-bin model in the formulation but use the convex hull description to generate cuts.
- This is called *lift and project*

Computational Results

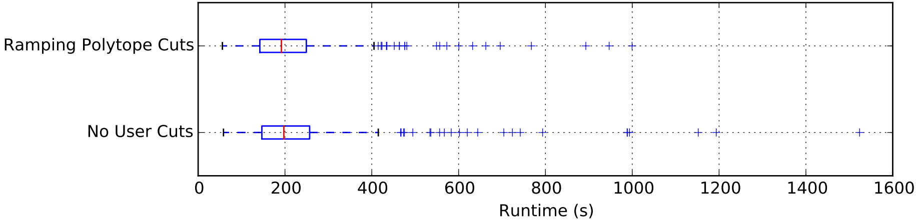
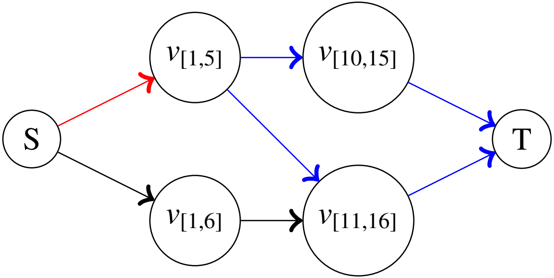


Figure: Computational Results for FERC Model (High Wind)

Identical Generators

- Sometimes there are identical generators in the UC problem.
- Unfortunately, we cannot aggregate all generators into the 3-bin model.
  - We can aggregate if there are no ramping constraints outside of startup and shutdown!
- However, we can easily account for additional identical generators in the extended formulation!
- Using the dynamic program context, this can be seen by performing multiple walks along the network.

A Picture



How Often Are There Identical Generators?

- Looking at a test case from California ISO (CAISO):
    - Of 610 generators, 465 are unique, giving a reduction of 20%!
    - Performing this aggregation solves problems 40% faster (from about 2 minutes to about 1 minute)!

How About Almost Identical?

- Consider the following 2 generators (actually in the CAISO set!):

Name	Min	Max	a	b	c
GEN 1264	11.1375	24.75	0.23224	2.71508	0.0003066
GEN 1477	11.43	25.4	0.23834	1.89607	0.0002988

- Are they really different?
- This happens a lot in the data!
- Solution: relax the data so they appear identical!

How About Almost Identical?

- Turn this:

Name	Min	Max	a	b	c
GEN 1264	11.1375	24.75	0.23224	2.71508	0.0003066
GEN 1477	11.43	25.4	0.23834	1.89607	0.0002988

- Into this:

Name	Min	Max	a	b	c
GEN 1264	11.1375	25.75	0.23224	1.89607	0.0002988
GEN 1477	11.1375	25.75	0.23224	1.89607	0.0002988

- To tighten things up a bit, we can add the constraint:

$$c_{1264+1477} \geq 1.89607 p_{1264+1477}$$
$$c_{1264+1477} \geq b + 2.71508 p_{1264+1477}$$

Results

- The data shows: There are a lot of almost identical generators.
- Aggregating near identical generators can reduce the number of generators from 610 to 315, for a 48% decrease (compared to 24% exactly identical).
- Solving the relaxed problems will be, we hope, much faster!
- The solutions are not always feasible, but they can be easily modified to become feasible.
- These modified solutions tend to be very close to the optimal solution (bases on limited tests, within 0.1%).

Computational Results

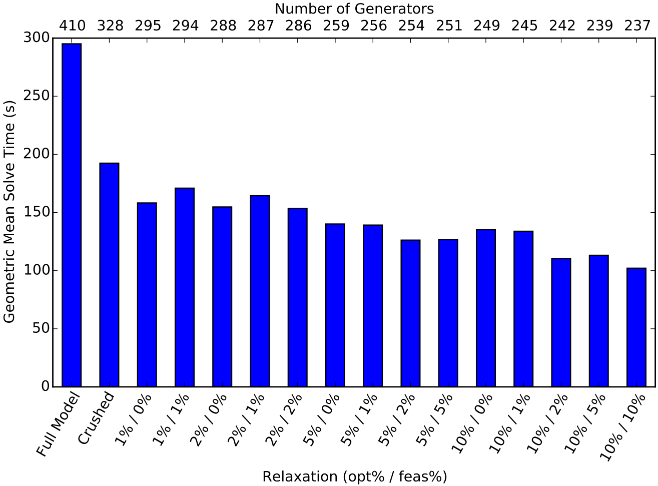


Figure: Computational Results for Almost Identical: CAISO

Future Work & Conclusions

Future Work

- We are in the process of trying to use the almost identical relaxation in order to find an exact optimal solution to the UC.
- To date we have ignored transmission. These ideas are still applicable even if generators are in different locations!

Conclusions

- We were able to come up with a compact convex hull description of a very important problem.
- This model allowed us to exploit the special structure in these Unit Commitment problems.