A Comparative Study of Stochastic and Security Constrained Unit Commitment Using High Performance Computing

ECC 2013, Zurich

Anthony Papavasiliou,
Department of Mathematical Engineering, CORE, UCL

Shmuel Oren,
IEOR Department, UC Berkeley

July 18, 2013
Outline

1 Introduction

2 Model
   - Unit Commitment Variants
   - Scenario Selection
   - Decomposition Algorithms

3 Results
   - System
   - Comparison of SUC and SCUC
Motivation and Research Objective

- Increased need for systematic approach to committing day-ahead reserves due to:
  - Renewable penetration
  - Demand response integration
- Four paradigms for systematic day-ahead scheduling:
  - Stochastic optimization
  - Security constrained optimization
  - Robust optimization
  - Probabilistically constrained optimization
- Our objective:
  - Compare relative performance of SUC and SCUC
  - Demonstrate benefits of parallel computation
Systematic Approaches to Unit Commitment

- Stochastic UC (Takriti 1996): minimize expected cost over weighted set of scenarios
  - Difficulty: scenario selection and probability assignment
  - Common solution approach: Lagrangian relaxation
- Security constrained UC: minimize no-contingency cost while withstanding failures without shedding load
  - (Wang 2008): exogenous reserve criteria, Benders
  - (Wu 2007): blend failures with scenarios, LR
- Robust UC (Jiang 2012, Bertsimas 2013): minimize cost of operation against worst-case uncertainty
  - Limited information about uncertainty required
  - Consistent with paradigm of system operators
- UC with probabilistic constraints (Ozturk 2004, Vrakopoulou 2013)
  - Limited information about uncertainty required
Parallel Computing Literature in Power Systems

- Monticelli et al. (1987): Benders decomposition algorithm for SCOPF
- Pereira et al. (1990): Applications of parallelization in various applications including SCOPF, composite (generator, transmission line) reliability, hydrothermal scheduling
- Falcao (1997): Survey of HPC applications in power systems
- Kim, Baldick (1997): Distributed OPF
- Bakirtzis, Biskas (2003) and Biskas et al. (2005): Distributed OPF
Industry practice for hydrothermal scheduling

The PSR architecture is depicted in the following diagram:

Since moving to the cloud, PSR has recorded impressive results. Mr. Pereira explains: "AWS is important to our consulting services in order to run our mathematical models in tolerable execution times, as well as for our customers when they buy our models to run them on their own. Internal measurements have been taken and the expected power of the cloud was proven to be the right direction. As an example, a glance of AWS usage in October 2010 revealed over 44,000 processor hours were carried out, which would have required 76 days to be handled using the local available..."
Unit Commitment Model

- Domain $\mathcal{D}$ represents min up/down times, ramping rates, thermal limits of lines, reserve requirements, import constraints

\[
(UC) : \min \sum_{g \in G} \sum_{t \in T} (K_g u_{gt} + S_g v_{gt} + C_g p_{gt})
\]
\[
s.t. \sum_{g \in G_n} p_{gt} = D_{nt}
\]
\[
P_g^- u_{gt} \leq p_{gt} \leq P_g^+ u_{gt}
\]
\[
e_{kt} = B_k (\theta_{nt} - \theta_{mt}), k = (m, n)
\]
\[
(p, e, u, v) \in \mathcal{D}
\]
Stochastic Unit Commitment Model

\[(SUC) : \min \sum_{g \in G} \sum_{s \in S} \sum_{t \in T} \pi_s (K_g u_{gst} + S_g v_{gst} + C_g p_{gst})\]

\[s.t. \sum_{g \in G_n} p_{gst} = D_{nst},\]

\[P_{gs}^- u_{gst} \leq p_{gst} \leq P_{gs}^+ u_{gst}\]

\[e_{kst} = B_{ks}(\theta_{nst} - \theta_{mst}), k = (m, n)\]

\[(p, e, u, v) \in D_s\]

\[u_{gst} = w_{gt}, v_{gst} = z_{gt}, g \in G_s\]

1. First stage: DA market realization for slow generators \(G_s\)
2. Renewable supply, line / generator outages
3. Second stage: RT market realization
Scenario-Based Security Constrained Unit Commitment

\[(SCUC) : \min \sum_{g \in G} \sum_{s \in S} \sum_{t \in T} \pi_s (K_g u_{gst} + S_g v_{gst} + C_g p_{gst})\]

s.t. \(\sum_{g \in G_n} p_{gst} = D_{nst}\)

\(P^-_{gs} u_{gst} \leq p_{gst} \leq P^+_{gs} u_{gst}, g \in G\)

\(e_{kst} = B_{ks}(\theta_{nst} - \theta_{mst}), k = (m, n)\)

\((p, e, u, v) \in D_s\)

\(\rho_{lst} = 0, l \in L, s \in S, t \in T\)

\(u_{gst} = w_{gt}, v_{gst} = z_{gt}, g \in G_s\)

- May not have feasible second-stage response
Scenario Selection [1 - 3]

- **Stochastic UC**: scenario selection algorithm inspired by importance sampling

  1. Generate a sample set $\Omega_S \subset \Omega$, where $M = |\Omega_S|$ is adequately large. Calculate the cost $C_D(\omega)$ of each sample $\omega \in \Omega_S$ against the best deterministic unit commitment policy and the average cost $\bar{C} = \frac{1}{M} \sum_{i=1}^{M} C_D(\omega_i)$.

  2. Choose $N$ scenarios from $\Omega_S$, where the probability of picking a scenario $\omega$ is $C_D(\omega)/(M\bar{C})$.

  3. Set $\pi_s = C_D(\omega)^{-1}$ for all $\omega^s \in \hat{\Omega}$.

- **Security Constrained UC**:

  1. $S$ is Cartesian product of renewable supply with no contingency and worst single-element contingencies.

  2. Equal $\pi_s > 0$ for no-contingency scenarios, $\pi_s = 0$ for single-element contingency scenarios.
Lagrange Relaxation for SUC [1 - 3]

\[ \mathcal{L} = \sum_{g \in G} \sum_{s \in S} \sum_{t \in T} \pi_s (K_g u_{gst} + S_g v_{gst} + C_g p_{gst}) \\
+ \sum_{g \in G_s} \sum_{s \in S} \sum_{t \in T} \pi_s (\mu_{gst}(u_{gst} - w_{gt}) + \nu_{gst}(v_{gst} - z_{gt})) \]
Motivation:
- Good feasibility cuts can be generated by severe contingencies
- Optimality cuts can be rapidly computed in parallel

Assumptions
- Convexity of value function: unit commitment has to be fixed in the first stage for all generators
- Ramping: assume away ramping constraints in order to decompose second-stage domain by time period, $D_{st}$
In order to avoid stall of standard feasibility cuts (Van-Slyke, Wets, 1969), pass entire set of power flow equations $D_{st}$ for most severe contingency.
130 units, 225 buses, 375 transmission lines
## Unit Characteristics

<table>
<thead>
<tr>
<th>Type</th>
<th>No. of units</th>
<th>Capacity (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuclear</td>
<td>2</td>
<td>4,499</td>
</tr>
<tr>
<td>Gas</td>
<td>94</td>
<td>20,595.6</td>
</tr>
<tr>
<td>Coal</td>
<td>6</td>
<td>285.9</td>
</tr>
<tr>
<td>Oil</td>
<td>5</td>
<td>252</td>
</tr>
<tr>
<td>Dual fuel</td>
<td>23</td>
<td>4,599</td>
</tr>
<tr>
<td>Import</td>
<td>22</td>
<td>12,691</td>
</tr>
<tr>
<td>Hydro</td>
<td>6</td>
<td>10,842</td>
</tr>
<tr>
<td>Biomass</td>
<td>3</td>
<td>558</td>
</tr>
<tr>
<td>Geothermal</td>
<td>2</td>
<td>1,193</td>
</tr>
<tr>
<td>Wind (deep)</td>
<td>10</td>
<td>14,143</td>
</tr>
<tr>
<td>Fast thermal</td>
<td>88</td>
<td>11,006.1</td>
</tr>
<tr>
<td>Slow thermal</td>
<td>42</td>
<td>19,225.4</td>
</tr>
</tbody>
</table>
Implementation

- Lawrence Livermore National Laboratory
  - 8 CPUs per node, 2.4 GHz and 10 GB per node
  - MPI calling on CPLEX Java callable library
- 30 scenarios:
  - SUC: importance sampling
  - SCUC: Cartesian product of ten renewable production scenarios with no-contingency case and two most severe contingencies (Diablo and San Onofre nuclear plants)
- 1,000 Monte Carlo outcomes
  - Spring weekdays (calibrated against NREL wind data)
  - 1% generator failure probability
  - 0.1% line failure probability
Conservative commitment of SCUC driven by assumption that all generators are slow.
Performance

Table: Daily cost breakdown ($)

<table>
<thead>
<tr>
<th></th>
<th>Startup</th>
<th>Min. load</th>
<th>Load shed</th>
<th>Fuel</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCUC</td>
<td>66.5</td>
<td>1,205.3</td>
<td>0</td>
<td>4,687.3</td>
<td>5,959.1</td>
</tr>
<tr>
<td>SUC</td>
<td>106.0</td>
<td>699.4</td>
<td>0.3</td>
<td>4,831.5</td>
<td>5,637.2</td>
</tr>
</tbody>
</table>

SCUC is more reliable, at the expense of a 5.4% cost increase.
Algorithm converged to optimal solution in 31 iterations. First feasible UC schedule detected in iteration 19.


Approach is not scalable as number of scenarios increases (due to growth of first-stage subproblem).
Algorithm ran for 80 iterations. Lower bound: $5.868M. Upper bound: $5.911M.

Marginal benefits vanish beyond 15 processors. Fully serial: 15.8 hours. Fully parallel: 47.7 minutes.

Approach is scalable as number of scenarios increases.
Conclusions and Perspectives

- **Tradeoffs:** The SCUC model achieves greater reliability at the expense of a 5.4% cost increase.

- **Parallelism:** Lagrange relaxation algorithm benefits more from parallelism.
  - Second-stage problems of SUC are more difficult to solve.
  - First-stage problem of SCUC is not decomposable.

- **Future work:** Improve feasibility cuts in Benders algorithm in order to scale for larger number of scenarios.
Thank you

Questions?

Contact: anthony.papavasiliou@uclouvain.be

References


Lagrangian Decomposition Algorithm

- Past work: (Takriti et al., 1996), (Carpentier et al., 1996), (Nowak and Römisch, 2000), (Shiina and Birge, 2004)
- Key idea: relax non-anticipativity constraints on both unit commitment and startup variables
  1. Balance size of subproblems
  2. Obtain lower and upper bounds at each iteration

Lagrangian:

\[
\mathcal{L} = \sum_{g \in G} \sum_{s \in S} \sum_{t \in T} \pi_s (K_g u_{gst} + S_g v_{gst} + C_g p_{gst}) \\
+ \sum_{g \in G_s} \sum_{s \in S} \sum_{t \in T} \pi_s (\mu_{gst} (u_{gst} - w_{gt}) + \nu_{gst} (v_{gst} - z_{gt}))
\]