Predict-and-Optimize Robust Unit Commitment

Paper Review & Improvement Proposal — Adithya Jaiswal, Aditi, Naman Meena, Swastic Keshari

What the Paper Does

The Unit Commitment problem is a daily scheduling task: decide which generators to switch on and at what output before the day begins, when demand is still uncertain. Traditional approaches predict first, optimize second — and those two steps never talk to each other, so forecast errors silently degrade the schedule.

This paper builds a joint framework where the forecasting model is trained knowing it will feed into a robust optimizer. Prediction errors directly shape the uncertainty set the scheduler must handle.

Three BiLSTM networks (M1: local, M2: federated, M3: sub-profile) produce separate demand forecasts, blended as:

$$\hat{U}(\mathbf{w}) = \sum_m w^{(m)}\hat{U}^{(m)}, \quad \mathbf{w}\in\mathcal{W},\ \sum_m w^{(m)}=1$$

The blended forecast builds an ellipsoidal uncertainty set $\mathcal{U}_1$ guaranteed — by Theorem 1 — to contain true demand with probability $\geq 1-\varepsilon$ over N samples:

$$\mathbb{P}^N\big[\Pr[u\in\mathcal{U}_1]\geq 1-\varepsilon\big]\geq 1-\delta$$

This set feeds a two-stage robust UC solved by C&CG. A small offline MLP surrogate (2 layers, 16 neurons, PCA to 3 components) approximates the optimal cost $I(\mathbf{w})$, so the daily best weights are found by a MILP instead of re-running the full UC. Theorem 2 tightens $\mathcal{U}_1\to\mathcal{U}_2$ using UC problem structure, reducing conservatism.

Algorithm — Step by Step

1. **Collect error samples.** Gather N_1 historical demand forecast errors across all buses and time slots.
2. **Train M1, M2, M3.** Run BiLSTM training for each forecaster independently; each gives a different $\hat{U}^{(m)}$.
3. **Build $\mathcal{U}_1$ via Theorem 1.** Fit an ellipsoid around the blended forecast that statistically covers true demand at confidence $1-\varepsilon$.
4. **Train MLP surrogate offline.** Solve full robust UC for N_2 weight samples; fit MLP on $(\hat{U}^{(m)},\mathbf{w})\to I^*$; encode into MILP coefficients.
5. **Daily: solve MILP for best w.** Use surrogate to pick the weight vector minimizing expected robust cost — takes seconds.
6. **Reconstruct $\mathcal{U}_2$ via Theorem 2.** Tighten the uncertainty set using UC constraints to cut physically impossible scenarios.
7. **Solve two-stage robust UC (eq. B).** Run C&CG on $\mathcal{U}_2$ to find day-ahead commitment x minimizing worst-case cost.
8. **Real-time re-dispatch.** Once demand is revealed, solve cheapest feasible $y\in\mathcal{Y}(x,u)$ without changing on/off decisions.

Pipeline: M1,M2,M3 $\xrightarrow{\text{blend}}$ $\hat{U}(\mathbf{w})$ $\xrightarrow{\text{Thm. 1}}$ $\mathcal{U}_1$ $\xrightarrow{\text{Thm. 2}}$ $\mathcal{U}_2$ $\xrightarrow{\text{C\&CG}}$ schedule x $\xrightarrow{\text{real-time}}$ re-dispatch y

Original Equations We Modify (see reverse for our changes)

(A) Chance-constrained problem — eq. 1 Foundation: find x and cost bound η such that real-time adjustment cost stays under η with prob. $\geq 1-\varepsilon$.

$$\min_{x\in\mathcal{X},\,\eta} f(x)+\eta \quad \text{s.t.} \quad \Pr\Big[\min_{y\in\mathcal{Y}(x,u)} h(y)\leq\eta\Big]\geq 1-\varepsilon$$

(B) Two-stage robust UC — eq. 2 ▷ Extended in Improvement II. Commit x day-ahead, re-dispatch y real-time against worst-case $u\in\mathcal{U}$.

$$\min_{x\in\mathcal{X}}\Big\{f(x)+\max_{u\in\mathcal{U}}\min_{y\in\mathcal{Y}(x,u)} h(y)\Big\}$$

(C) Surrogate training loss — offline ▷ Modified in Improvement I. Train MLP once on all history, flat weights, frozen forever.

$$\mathcal{L}(\theta)=\sum_{n=1}^{N_2}\big(I_n-\hat{I}(x_n;\theta)\big)^2$$

(D) Daily weight MILP — eq. 6 ▷ Extended in Improvement III. Pick $\mathbf{w}$ minimising surrogate cost. Only constraints: $w\geq 0$, sum to 1. Nothing stops day-to-day jumps.

$$\min_{\mathbf{w},\hat{U},s,v,z,I} I \quad \text{s.t.} \quad (5),\ w\geq 0,\ \sum_m w^{(m)}=1$$

Our Three Improvements

For each: gap in the original paper → original eq. → our fix

IMPROVEMENT I
Online Adaptive Surrogate
modifies eq. (C)

The MLP in eq. (C) is trained once and never touched again. Real systems don't stay still — load shifts with the season, new generators come online, federated nodes in M2 drop out. A frozen model slowly loses touch with current conditions. The paper tests on just one day, so none of this drift shows up in their numbers.

Original eq. (C):

Train θ once on $\mathcal{D}_{\text{hist}}$.
$\mathcal{L}(\theta) = \sum_n (I_n - \hat{I}(x_n;\theta))^2$
Same θ used every future day.

Our eq. (C′):

Rolling window: $\mathcal{D}_t = \{(x_n, I_n)\}_{n=t-W}^{t}$

$$\mathcal{L}(\theta_t) = \sum_{n=t-W}^{t} \lambda^{t-n}(I_n - \hat{I})^2 + \mu\|\theta_t - \theta_{t-1}\|^2$$

K steps/day. λ=0.95, μ=0.01
W=30, K=30

λ^{t-n}: older data counts less but isn't discarded.

$\mu\|\theta_t - \theta_{t-1}\|^2$: prevents the model wiping out what it knew before each update.

After each daily update, MILP coefficients W_{ij}, B_{ij} are refreshed from the new θ_t. Theorems 1 & 2 don't depend on θ — they're untouched.

IMPROVEMENT II
Three-Stage Robust UC
modifies eq. (B)

Eq. (B) gives operators two moments to act: the night-before schedule and real-time re-dispatch. In practice, there's a third window 4–6 hours before real-time where forecast error drops from 30–40% to 10–15%. The two-stage model can't use it — all reserves are locked at midnight. That means holding expensive standby capacity for 24 hours based on a forecast that's already been improved by morning. The paper flags this gap itself as future work.

Original eq. (B):

$$\min_x \Big\{ f(x) + \max_{u\in\mathcal{U}} \min_{y\in\mathcal{Y}(x,u)} h(y) \Big\}$$

Two stages only: x then y.

Our eq. (B′):

$$\min_{x\in\mathcal{X}} f(x) + \max_{u^{(1)}\in\mathcal{U}^{(1)}} \min_{z\in\mathcal{Z}(x,u^{(1)})} g(z) + \max_{u^{(2)}\in\mathcal{U}^{(2)}(z)} \min_{y\in\mathcal{Y}(x,z,u^{(2)})} h(y)$$

$z \in \mathcal{Z}(x, u^{(1)})$: intra-day reserve shift at 4–6 hr — can move reserves around but *cannot* change on/off decisions.

$g(z)$: cost of that mid-day adjustment.

Theorem 1 applied twice gives two independent uncertainty sets $\mathcal{U}^{(1)}, \mathcal{U}^{(2)}$ each with its own guarantee:

$$\mathbb{P}^N\big[\Pr[u^{(s)} \in \mathcal{U}^{(s)}] \geq 1-\varepsilon_s\big] \geq 1-\delta_s$$

Solved by nested C&CG: outer loop over $u^{(1)}$, inner over $(z, u^{(2)}, y)$. Theorem 2 reconstruction extends to Stage 3.

IMPROVEMENT III
Time-Varying Weight Stability
modifies eq. (D)

Eq. (D) picks a fresh $\mathbf{w}$ every morning with no limit on how much it can change. The MILP might output $(0.9, 0.05, 0.05)$ on Monday and $(0.02, 0.95, 0.03)$ on Tuesday — not because any forecaster got better or worse overnight, just because the solver found a different corner of the feasible region. Those swings destabilize $\hat{U}(\mathbf{w})$ and cascade into the uncertainty set. Also, eq. (C) weights a 2-year-old sample the same as last week's, which makes little sense for a system that evolves gradually.

Original eq. (D):

$\min_{\mathbf{w},\ldots} I$ s.t. (5), $w \geq 0$,
$\sum_m w^{(m)}$=1
No cap on $\|w_t - w_{t-1}\|$.
All history weighted equally.

Our eq. (D′):

$\min_{\mathbf{w},\ldots} I$ s.t. (5), $w \geq 0$,
$\sum_m w^{(m)}$=1,
$\|w_t - w_{t-1}\|_1 \leq \Delta_w$

Linearised — no new binaries:
$d_m \geq w_t^{(m)} - w_{t-1}^{(m)}$
$d_m \geq w_{t-1}^{(m)} - w_t^{(m)}$, $\forall m$
$\sum_m d_m \leq \Delta_w$, Δ_w=0.2

Recency-weighted loss (from C′):
$\mathcal{L} = \sum_n \lambda^{N-n}(I_n - \hat{I}_n)^2$, λ=0.97

The ℓ_1 constraint caps total daily weight shift at 0.2. Linearised with M extra continuous variables d_m and $2M$ linear constraints — MILP complexity barely changes, no new binaries added.

Recency-biased loss mirrors eq. (C′): older samples discounted by λ^{N-n}. Theorems 1 & 2 unaffected.