Bayesian Inference for Estimating Generation Costs in Electricity Markets

Matthias Pirlet, Adrien Bolland, Alexandre Huynen, Quentin Louveaux, Gilles Louppe, Damien Ernst

University of Liège - Engie

PSCC- Limassol, 11 June 2026

LIÈGE université

ENGIE

What do we observe, what is hidden ?

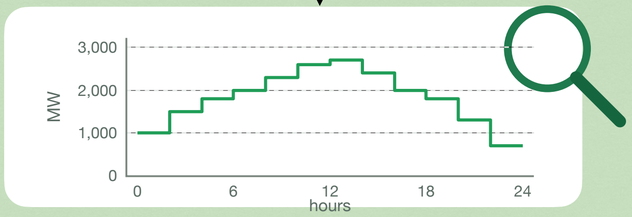
Observed

G1

G2

G3

produces



What generation costs explain it ?

Hidden

Marginal cost c

Start-up cost s

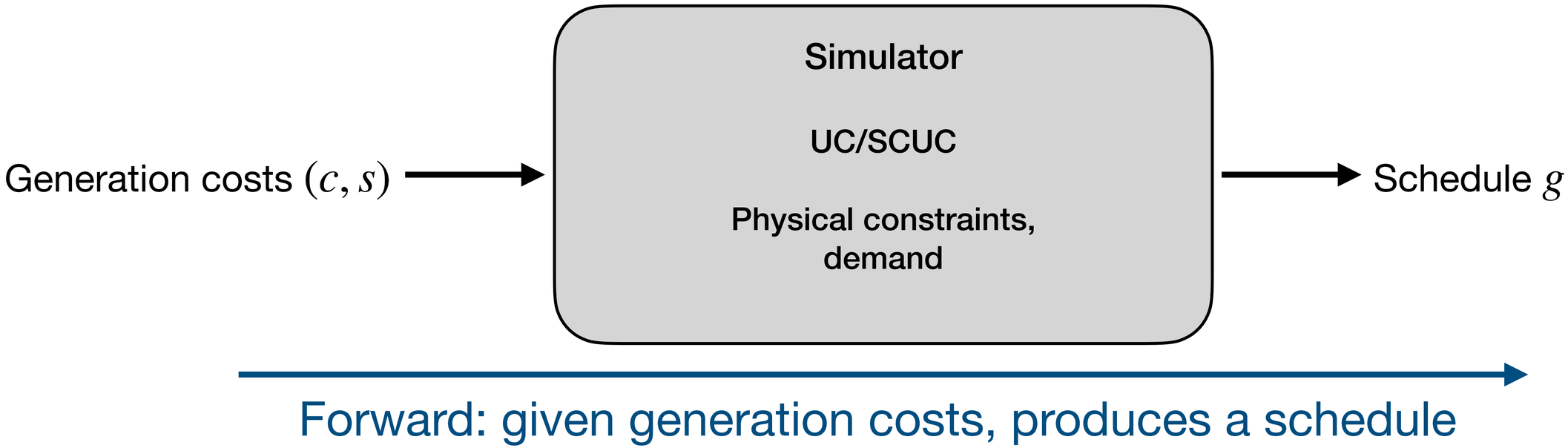
?

Market monitoring

Price forecasting

Investment valuation

From costs to schedules



Forward model

Latent Variable Model (LVM)

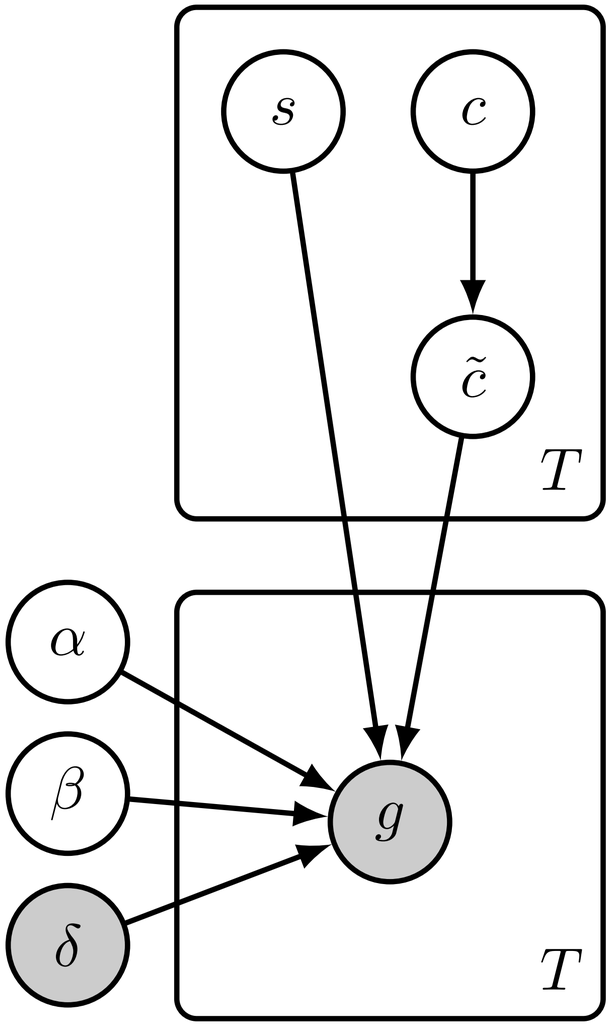
Build a mathematical model that reflects the behavior of the competitors.

Challenge - schedule g we observe depends on some hidden variables

- c marginal cost,
- s startup cost,
- $\tilde{c}$ opportunity cost,
- α availability of a unit,
- β availability of a line.

It also depends on known and observed variables

- technical limits,
- δ demand.



$$c \sim \mathcal{U}(\cdot, \cdot)$$

$$s \sim \mathcal{U}(\cdot, \cdot)$$

$$\tilde{c} \sim \mathcal{N}(c, \sigma^2)$$

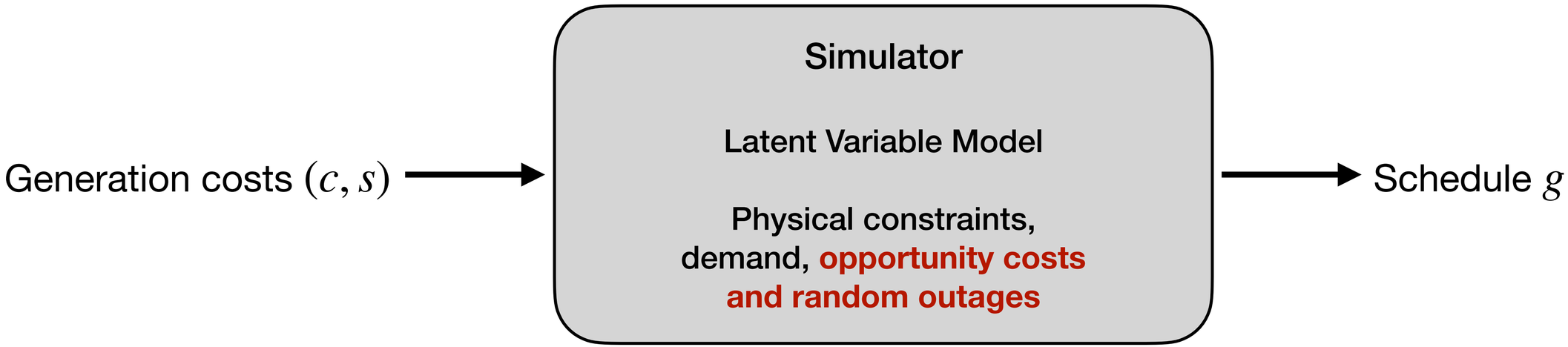
$$\alpha \sim \text{Bernoulli}(\cdot)$$

$$\beta \sim \text{Bernoulli}(\cdot)$$

$$\delta \sim p(\delta)$$

$$g = \text{UC}(\tilde{c}, s, \alpha, \beta, \delta)$$

From costs to schedules… and back



Forward: given generation costs, produces a schedule

Inverse: given the observed schedule, infer the costs that could have produced it

Bayesian inference

- A prior model that can sample parameters $\theta_i \sim p(\theta)$ where $\theta = \{c, s\}$.
  - Expert knowledge on the parameters.
- A forward model i.e., a simulator that can generate samples, $x_i \sim p(x \mid \theta_i)$.
  - Mapping from costs to schedule.
- Target: Posterior $p(\theta \mid x_{obs})$.
  - Which costs are compatible with the observed schedule.

$$p(\theta \mid x_{obs}) = \frac{p(x_{obs} \mid \theta)\ p(\theta)}{p(x_{obs})}$$

Challenge: simulator is stochastic. Likelihood cannot be computed. We can only sample from it.

Balanced Neural Posterior Estimation (BNPE)

Amortized inference

Train a neural network $q_\phi(\theta \mid x)$ mapping observations to a posterior over costs.

Objective: maximize the expected log-posterior under the joint distribution $p(\theta, x)$ which is equivalent to

$$\min_{\phi} \mathop{\mathbb{E}}_{p(\theta,x)} \Big[-\log q_\phi(\theta \mid x) \Big],$$

Once trained, inference for any new observation is a single forward pass. This is known as amortized inference.

Inverse optimization

Polar cone method

Recover costs for which observed schedule x^{obs} is ϵ-optimal.

$$\theta^{\top}(g - g^{\text{obs}}) \geq -\epsilon \quad \forall g \in \mathcal{G}(\delta_{\text{obs}}),$$

- The feasible cost set (polar cone) is approximated by iterative refinement until this ϵ-optimality condition holds.
- Deterministic only. It ignores stochasticity of our simulator.
- Training-free, adapts directly to new models **but** point estimate, no confidence interval.

Setup and validation

Setup

- IEEE RTS-96 (2016): 24 buses, 12 generators, 38 lines, 24 hours (hourly).
- Wide uniform priors: $c \sim \mathcal{U}(10,50)$ €/MWh, $s \sim \mathcal{U}(500,10\,000)$€.
- Observation is synthetic i.e., generated from a known nominal value.

What validation can show

- We never observe the posterior, so the error cannot be measured directly.
- Diagnostics can reveal a wrong posterior but can never certify a correct one.

Results

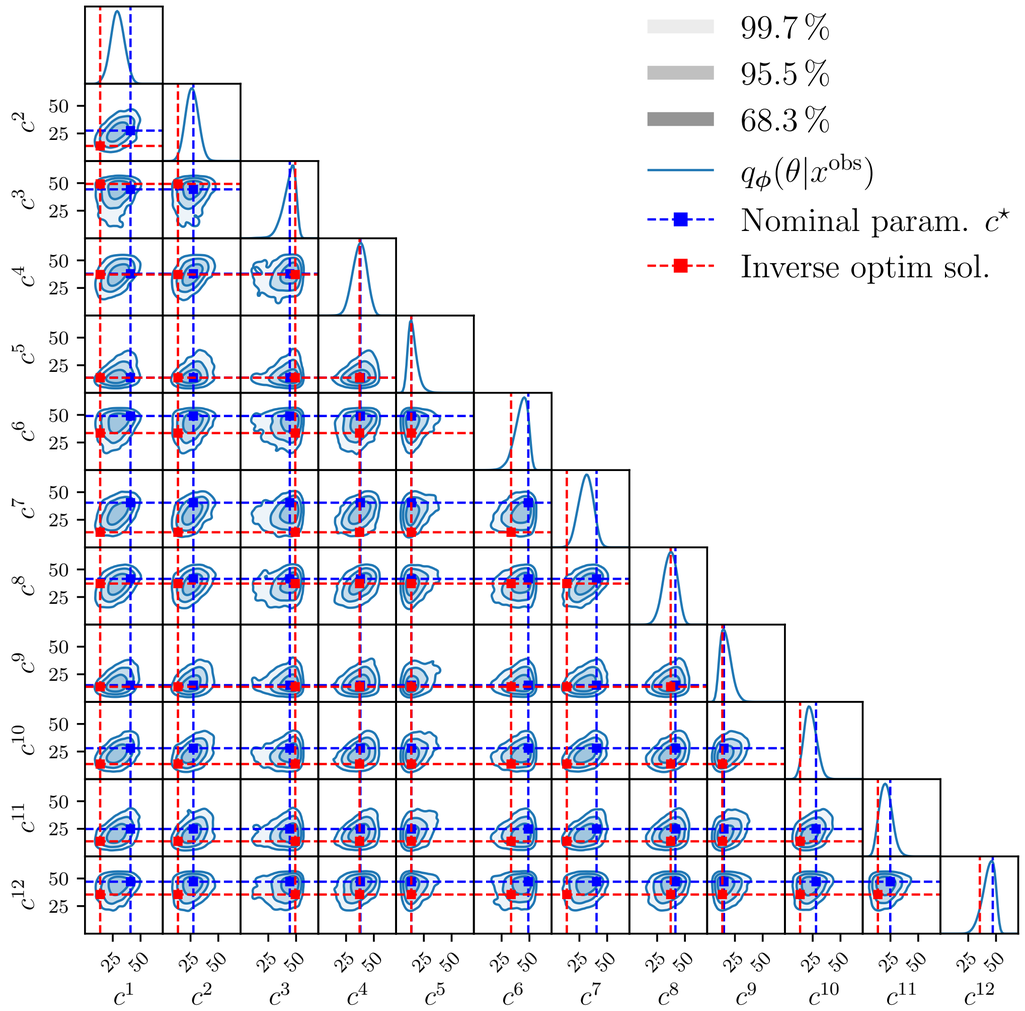
Can we recover marginal costs ?

Marginal costs

Posterior distributions are narrow, peaked and far from the $\mathcal{U}(10,50)$ prior distribution.

Nominal values lie in high-density regions.

The model extracts information from observed schedules and represents dependencies between generators.



Results

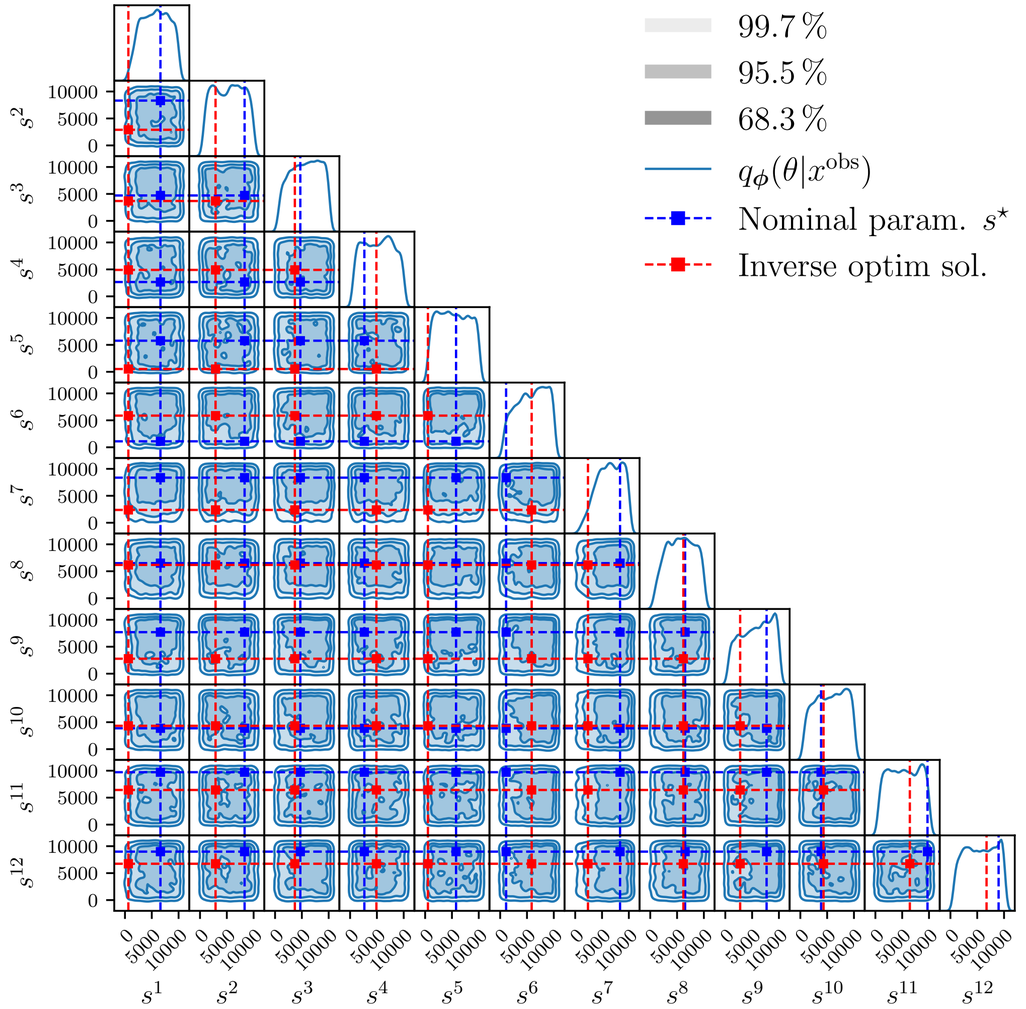
Can we recover start-up costs ?

Start-up costs

Posterior distributions remain close to the $\mathcal{U}(500, 10\,000)$ prior.

Observed schedules contain little information about startup costs (only enters in the infrequent commitment decisions).

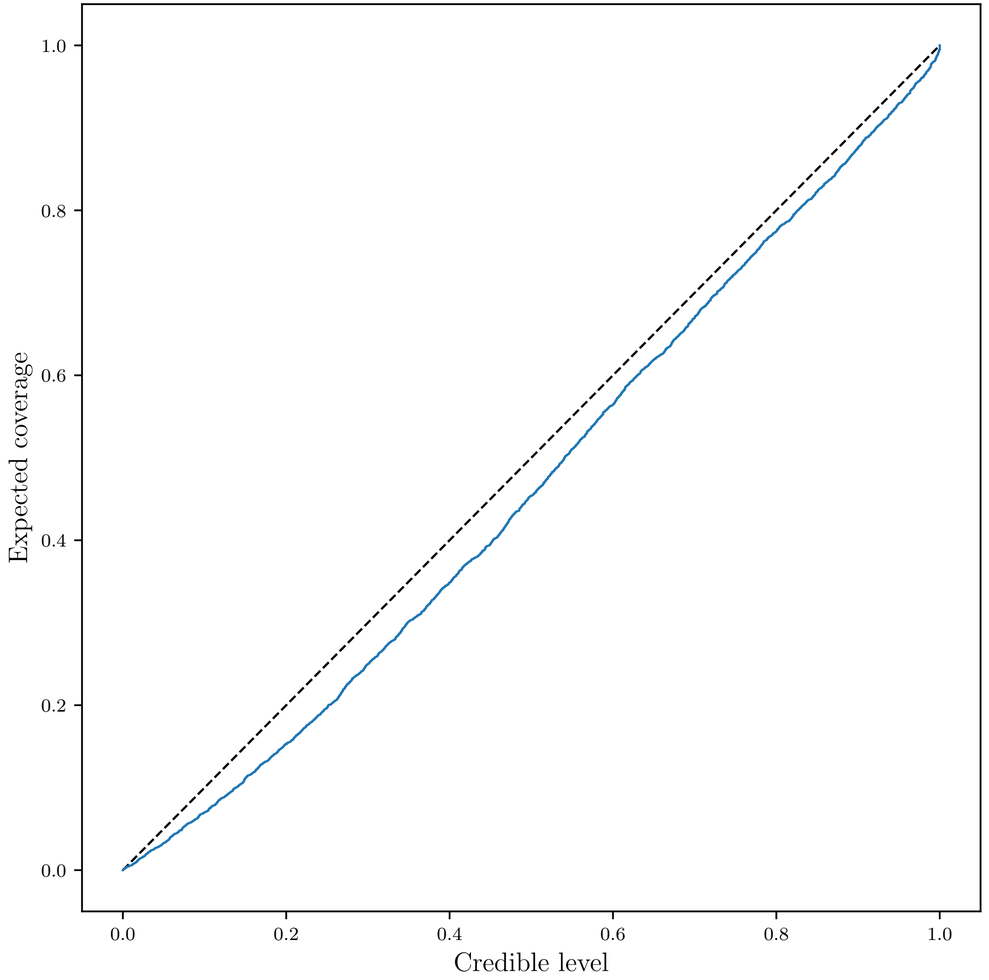
The method reflects this correctly: it returns a near-prior posterior instead of an arbitrary value.



Results

Can we trust the uncertainty ?

- This plot assesses the calibration of our learned posterior distributions i.e., how well our predicted uncertainty ranges reflect the true uncertainty
- Our results indicate a very small overconfidence.
- May be due to high dimensional parameter space for the algorithm used (normalizing flows).



Takeaway

- A latent variable model of the market.
  - Cost estimation formulated as Bayesian inference, capturing stochasticity.
- BNPE
  - Well-calibrated approximate posterior and instantaneous inference for any new schedule.
- Inverse optimization
  - Training-free that adapts to model changes but deterministic and point estimate.

Thank you!

What is Unit Commitment (UC)?

UC is a fundamental optimization problem in power systems.

Objective: Determine the optimal on/off status and output level for each power plant over time, to meet forecasted electricity demand at minimum cost.

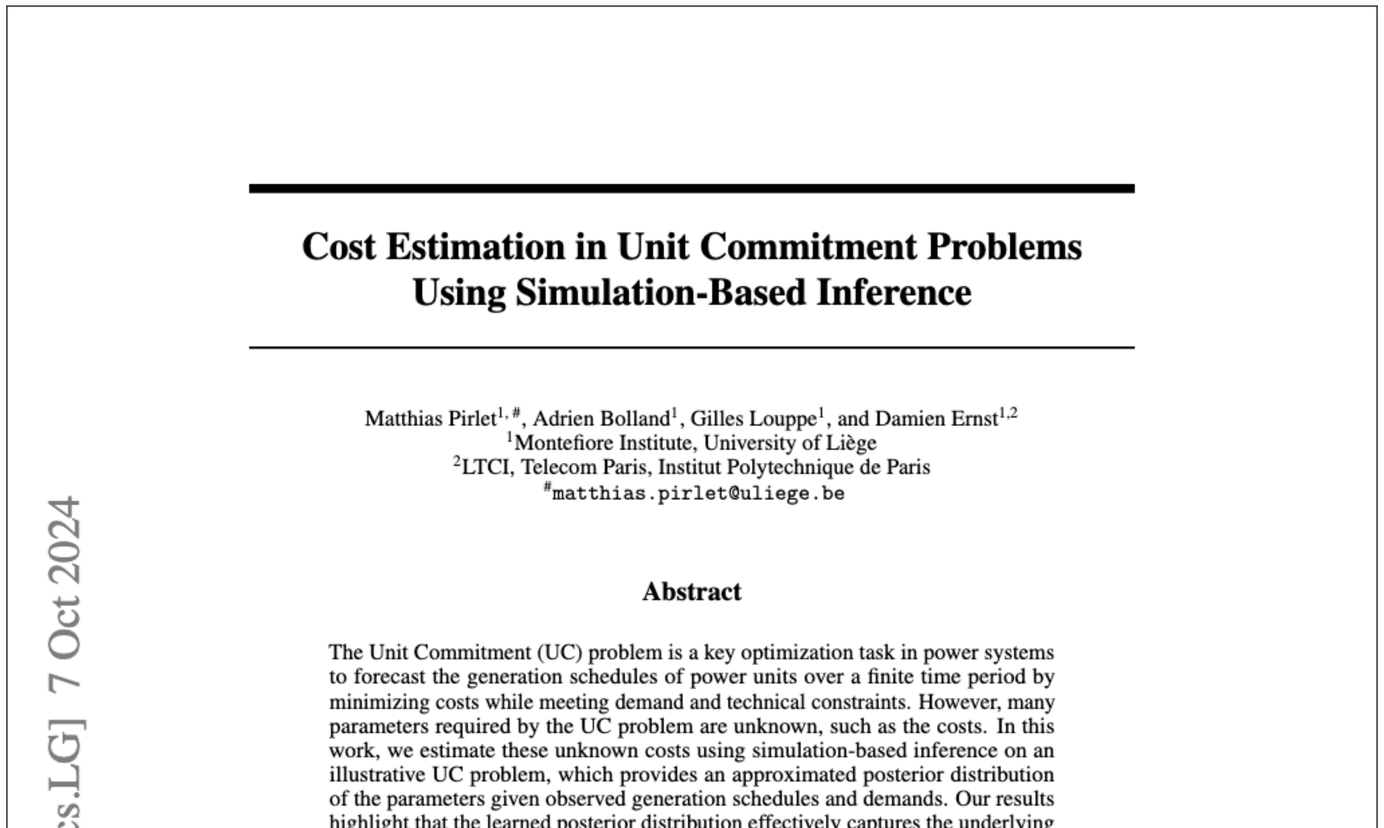
This family of problems involves economic objectives with cost parameters (e.g., generation/start-up costs) subject to technical constraints (e.g., generator ramp rates, minimum up/down times, capacity limits).

Output: Optimal generation schedule x.

First paper

Cost Estimation in Unit Commitment Problems Using Simulation-Based Inference

Accepted at D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers Workshop @ NeurIPS



cs.LG] 7 Oct 2024

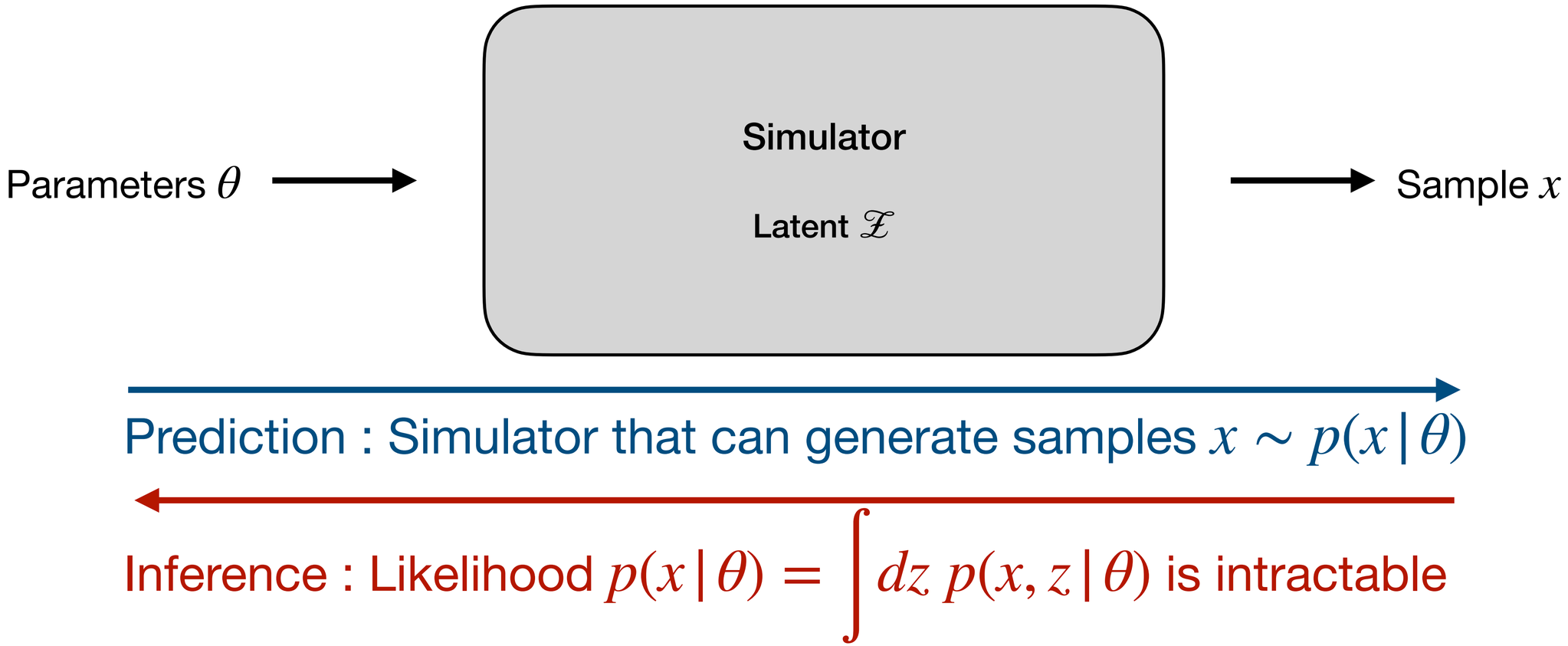
Cost Estimation in Unit Commitment Problems Using Simulation-Based Inference

Matthias Pirlet[1,#], Adrien Bolland[1], Gilles Louppe[1], and Damien Ernst[1,2]
[1]Montefiore Institute, University of Liège
[2]LTCI, Telecom Paris, Institut Polytechnique de Paris
[#]matthias.pirlet@uliege.be

Abstract

The Unit Commitment (UC) problem is a key optimization task in power systems to forecast the generation schedules of power units over a finite time period by minimizing costs while meeting demand and technical constraints. However, many parameters required by the UC problem are unknown, such as the costs. In this work, we estimate these unknown costs using simulation-based inference on an illustrative UC problem, which provides an approximated posterior distribution of the parameters given observed generation schedules and demands. Our results highlight that the learned posterior distribution effectively captures the underlying

Simulation-Based Inference (SBI)



Problem statement

Latent Variable Model (LVM)

Why LVM ?

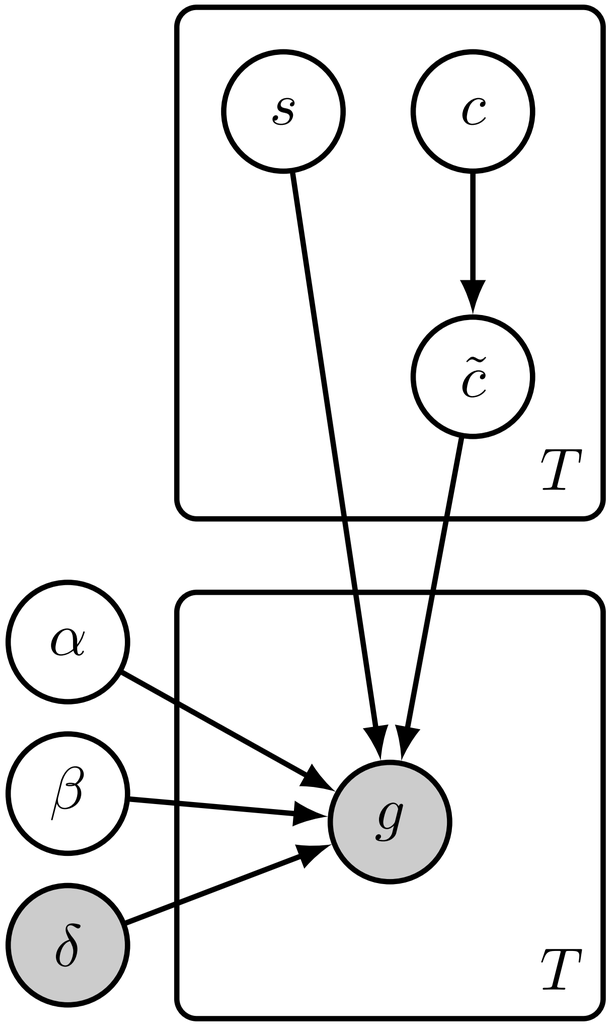
Competitors decision drivers are *unobserved*, but the resulting dispatch is *observed*.

This naturally creates an inverse problem

> Given the observed dispatch, infer the hidden parameters that generated it.

The LVM provides
- forward model of how dispatch is produced,
- way to represent latent (unknown) parameters,
- way to quantify uncertainty around the inferred parameters.

In short
- Forward = LVM.
- Inverse = parameter inference from observed dispatch.



$$c \sim \mathcal{U}(\cdot, \cdot)$$
$$s \sim \mathcal{U}(\cdot, \cdot)$$
$$\tilde{c} \sim \mathcal{N}(c, \sigma^2)$$

$$\alpha \sim \text{Bernoulli}(\cdot)$$
$$\beta \sim \text{Bernoulli}(\cdot)$$
$$\delta \sim p(\delta)$$
$$g = \text{UC}(\tilde{c}, s, \alpha, \beta, \delta)$$

Balanced Neural Posterior Estimation (BNPE)

Amortized inference

Among various SBI methods, we employ BNPE, which trains a neural network to learn the posterior distribution $p(\theta \mid x)$ by directly mapping observations to parameters.

We denote our model as $q_\phi(\theta \mid x)$, where ϕ are the neural network parameters.

This is achieved by maximizing the expected log-likelihood of under the true joint distribution $p(\theta, x)$ which is equivalent to

$$\min_{\phi} \mathop{\mathbb{E}}_{p(x)} \Big[\mathrm{KL}\big(p(\theta \mid x) \parallel q_\phi(\theta \mid x) \big) \Big],$$

Once the network is trained on a simulated dataset, it can rapidly infer the posterior distribution $p(\theta \mid x)$ for any new observation. This is known as amortized inference.

Inverse optimization

Polar cone method

An observed schedule x^{obs} is optimal for the UC problem with any parameter θ that satisfies

$$\theta^{\top}(x - x^{\text{obs}}) \geq 0 \quad \forall x \in \mathcal{X},$$

The set of parameters that satisfy this equation is denoted as the polar cone $\Theta^{\star}$.

Finding such a parameter would require enumerating infinitely many constraints. The method here involves the iterative refinement of an approximation of this polar cone.

Inverse optimization

Polar cone method

You start with an hypercube $\Theta^{(0)} = [\theta^{\min}, \theta^{\max}]$,

At iteration k, you sample a parameter θ^{ref} in this hypercube that you project onto the current approximation of polar cone $\Theta^{(k)}$

$$\min_{\theta \in \Theta^{(k)}} \|\theta - \theta^{\text{ref}}\|$$

You solve the UC with these costs to obtain schedule $x^{(k)}$ and check the optimality condition.

Inverse optimization

Polar cone method

- If $\theta^{(k)\top}(x^{(k)} - x^{\text{obs}}) = 0$, x^{obs} is optimal for the current cost parameters, the algorithm has converged.

- If $\theta^{(k)\top}(x^{(k)} - x^{\text{obs}}) < 0$, then $\theta^{(k)}$ violates the optimality condition and we refine polar cone approximation

$$\Theta^{(k+1)} = \{\theta \in \Theta^{(k)} : \theta^{\top}(x^{(k)} - x^{\text{obs}}) \geq 0\}$$

In practice, we stop when $\theta^{(k)\top}(x^{(k)} - x^{\text{obs}}) > -\epsilon$, meaning x^{obs} is ϵ-optimal, and parameter $\theta^{(k)}$ serves as an approximate solution.

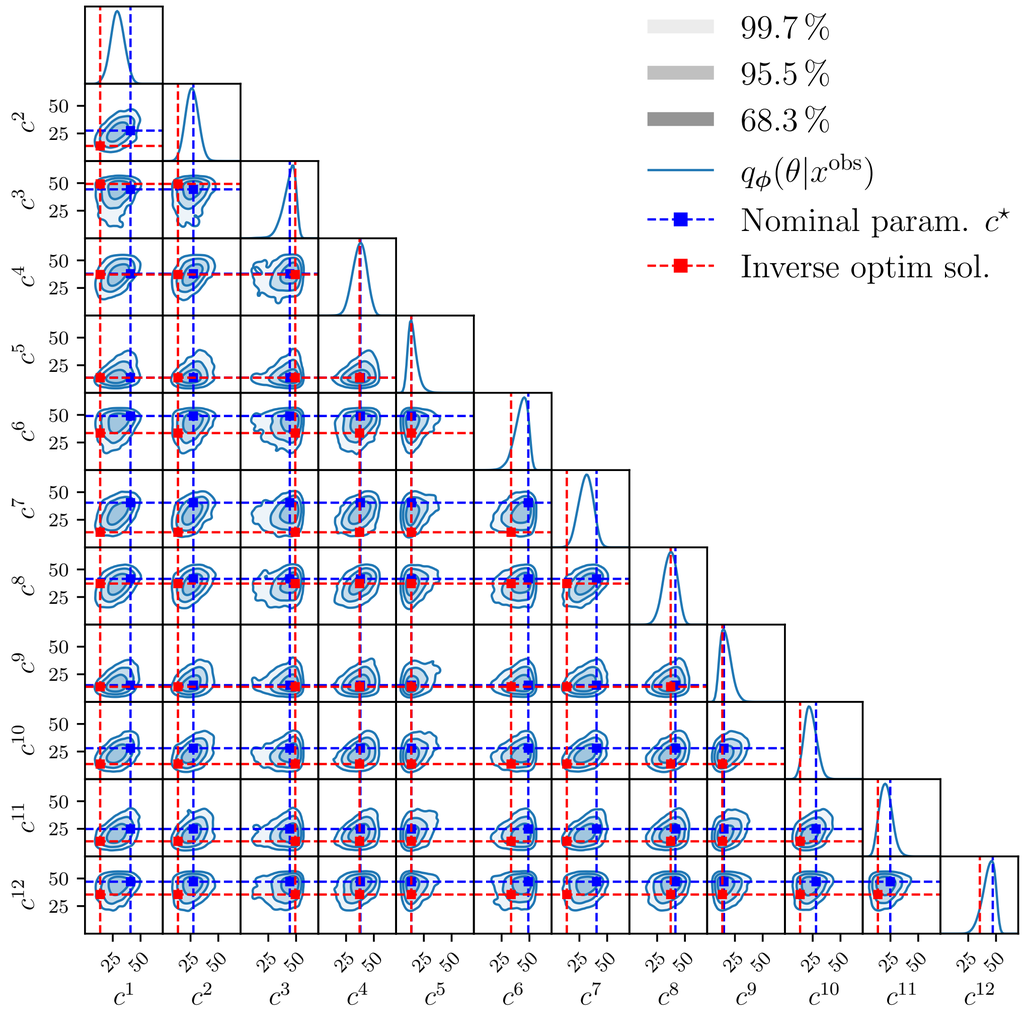
This method is only applicable for inverting deterministic models. We therefore ignore stochastic random bidding and model outages. Despite this limitation, inverse optimization provides fast point estimates for time-critical applications.

Results

BNPE correlation plot

This corner plot visualizes the posterior probability distributions given a single observed generation schedule.

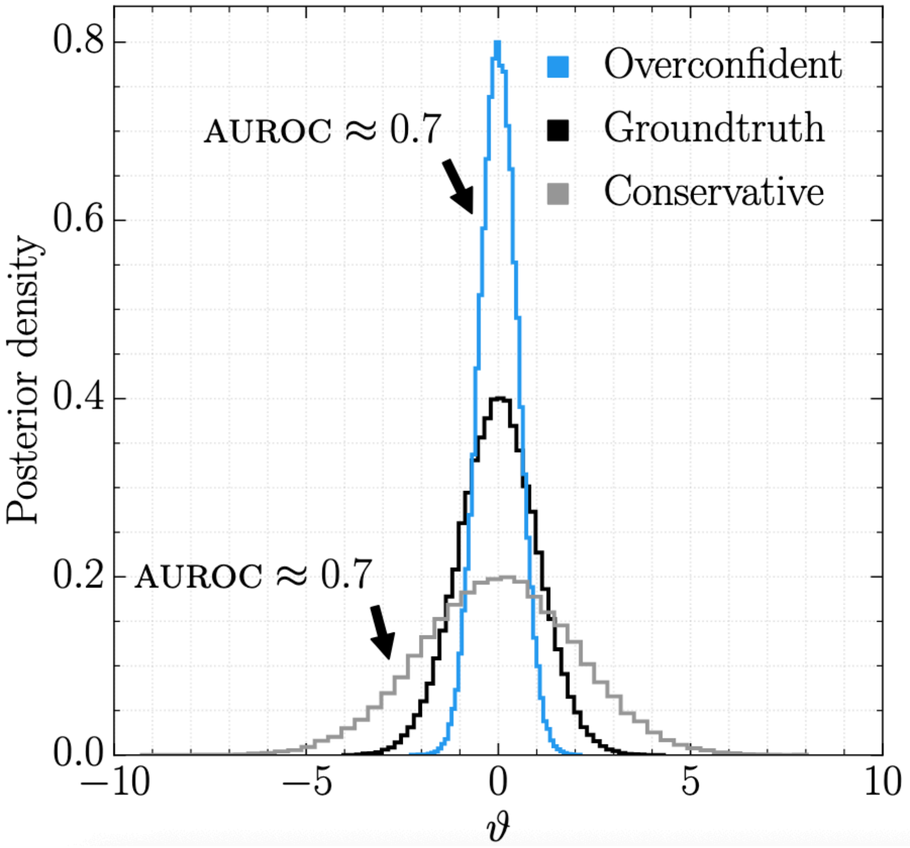
- Diagonal: marginal distributions for individual parameters.
- Off-diagonal: joint distribution between 2 parameters;
- Blue dot: nominal cost parameter used to generate the observed data.
- Red dot: solution of the inverse optimization problem.



Results

Coverage plot

- This plot assesses the calibration of our learned posterior distributions i.e., how well our predicted uncertainty ranges reflect the true uncertainty
- For example, if we claim a 90% credible interval for a parameter, we expect the true parameter to fall within that interval 90% of the time across many trials.



Sanity check

Validating our inferred parameters

We perform a posterior predictive as a self-consistency diagnostic for the learned posterior

$$p(x \mid x_{obs}) = \int_{\theta} p(x \mid \theta)\ q_{\phi}(\theta \mid x_{obs})\ d\theta$$

Where:

- $p(x \mid \theta)$ is our simulator (LVM);
- $q_{\phi}(\theta \mid x_{obs})$ is our BNPE estimator from which we can sample.

Sanity check (1)

Posterior predictive distribution

