

Market_1



An energy company is interested in investing in wind-power production. To assess the viability of this investment, the company wants to evaluate the expected revenue of a wind-power producer in a competitive spot market.

Suppose there are five energy producers participating in the day-ahead spot market. Three of them use conventional sources for energy production, such as coal, while the other two producers use solar and wind power, respectively. The market is cleared once per day for 24 hourly intervals of the following day. For each dispatch interval $t=1, \dots, 24$, every producer submits a bid (b_t, p_t) specifying the energy quantity b_t (MWh) it is willing to supply and the corresponding minimum acceptable price p_t (per MWh). Then, the market operator follows a merit-order process to determine the market-clearing price c_t and dispatch schedule, ensuring the total market demand D_t is met, where $D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$ with constant mean and variance. Merit-order ensures that any bid (b_t, p_t) at interval t , if $p_t \leq c_t$, the bid is accepted, and the producer is committed to delivering b_t at c_t . Otherwise, the producer earns zero revenue.

From the perspective of the wind-power producer, the bid (b_{it}, p_{it}) from a conventional producer i can be modeled as follows: b_{it} can be assumed to be stable, while $p_{it} \sim \mathcal{N}(\mu_{pi}, \sigma_{pi}^2)$ with deeply uncertain mean and variance since its bidding strategy is unknown. For the solar-power producer, its b_{st} is associated with the time t and assumed to be $b_{st} = \max(0, a + b \cos(\frac{2\pi t}{24}))$, where a and b can be forecast based on historical data thereby assumed to be constants. Its p_{st} is also uncertain, where $p_{st} \sim \mathcal{N}(\mu_{ps}, \sigma_{ps}^2)$ with deeply uncertain mean and variance.

The actual energy production of the wind-power producer G_t is also uncertain, as it depends on weather conditions. The production G_t can be forecast from historical data and is modeled as $G_t \sim \mathcal{N}(\mu_G, \sigma_G^2)$ with constant mean and variance. In this case, the actual production may differ from the quantity committed to dispatch one day in advance. If the producer under-delivers relative to its dispatched quantity, the shortfall incurs a penalty of q_u per MWh. This cost is imposed by the market operator to maintain the grid's safe operational conditions and to ensure supply-demand balance in real time.

The energy company aims to build a simulation model to evaluate the expected revenue of the wind-power producer on any future day by selecting the hourly bid quantity and price (b_{wt}, p_{wt}) , while accounting for uncertain generation and stochastic market clearing prices. This model can then be used to explore outcomes across various short-term and long-term scenarios, thereby informing the viability of the investment.

Please formalize this problem description as a simulation model and specify the key components, including state variables, decision variables, transition functions, objective functions, stochastic variables, constant parameters and deep uncertainties.

Below is a clean **formalization of the simulation model** from the wind-power producer's perspective. The structure is suitable for Monte-Carlo simulation and later extension to optimization or robust

analysis.

1. Time Structure and Market Setting

- **Time horizon:** One day, discretized into hourly intervals

$$t = 1, \dots, 24$$

- **Market:** Competitive day-ahead spot market with **merit-order clearing** and **uniform pricing**.
- **Participants:**
 - 3 conventional producers $i \in \mathcal{C} = \{1, 2, 3\}$
 - 1 solar producer s
 - 1 wind producer w (the decision maker)

2. State Variables

At each hour t , the system state consists of exogenous realizations:

$$S_t = (D_t, G_t, \{p_{it}\}_{i \in \mathcal{C}}, p_{st})$$

where:

- D_t : total market demand
- G_t : realized wind generation
- p_{it} : bid prices of conventional producers
- p_{st} : bid price of solar producer

The state is **fully stochastic** and independent across hours.

3. Decision Variables (Wind Producer)

For each hour t , the wind producer chooses:

$$(b_{wt}, p_{wt})$$

where:

- $b_{wt} \geq 0$: quantity offered (MWh)
- $p_{wt} \geq 0$: minimum acceptable price (€/MWh)

These are chosen **day-ahead**, before uncertainties are realized.

4. Deterministic Parameters

Market and Penalties

- $q_u > 0$: penalty per MWh of under-delivery

Solar Generation Profile

$$b_{st} = \max \left(0, a + b \cos \left(\frac{2\pi t}{24} \right) \right)$$

with constants a, b .

Conventional Producer Quantities

- $b_{it} = \bar{b}_i$, constant and known

5. Stochastic Variables (Aleatory Uncertainty)

Market Demand

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$$

Wind Generation

$$G_t \sim \mathcal{N}(\mu_G, \sigma_G^2)$$

Conventional Producer Bid Prices

$$p_{it} \sim \mathcal{N}(\mu_{pi}, \sigma_{pi}^2), \quad i \in \mathcal{C}$$

Solar Producer Bid Price

$$p_{st} \sim \mathcal{N}(\mu_{ps}, \sigma_{ps}^2)$$

All stochastic variables are assumed **independent across producers and time**, unless extended later.

6. Deep Uncertainties (Epistemic Uncertainty)

The following parameters are **poorly known** and treated as scenario-dependent inputs:

- μ_{pi}, σ_{pi}^2 : conventional bidding behavior
- μ_{ps}, σ_{ps}^2 : solar bidding behavior

These define **outer-loop scenarios** in the simulation (e.g., strategic vs aggressive bidding).

7. Market Clearing Mechanism (Transition Function)

Merit-Order Clearing

For each hour t :

1. Collect all bids:

$$\mathcal{B}_t = \{(b_{it}, p_{it}), (b_{st}, p_{st}), (b_{wt}, p_{wt})\}$$

2. Sort bids by increasing price.
3. Accept bids sequentially until total accepted quantity meets D_t .
4. Market-clearing price:

$$c_t = \max\{p : (b, p) \text{ is accepted}\}$$

8. Wind Producer Dispatch and Imbalance

Accepted Quantity

$$x_{wt} = \begin{cases} b_{wt}, & p_{wt} \leq c_t \\ 0, & p_{wt} > c_t \end{cases}$$

Under-Delivery

$$u_t = \max(0, x_{wt} - G_t)$$

9. Revenue Function

Hourly Revenue

$$R_t = c_t \cdot x_{wt} - q_u \cdot u_t$$

Daily Revenue

$$R = \sum_{t=1}^{24} R_t$$

10. Objective Function

The wind producer seeks to maximize **expected daily revenue**:

$$\max_{\{b_{wt}, p_{wt}\}_{t=1}^{24}} \mathbb{E}[R] = \mathbb{E} \left[\sum_{t=1}^{24} (c_t x_{wt} - q_u \max(0, x_{wt} - G_t)) \right]$$

Expectation is taken over:

$$(D_t, G_t, p_{it}, p_{st})$$

11. Simulation Architecture

Inner Loop (Monte Carlo)

- Sample stochastic variables
- Clear market
- Compute revenues

Outer Loop (Scenarios)

- Vary deep uncertainty parameters
 - Stress-test bidding strategies
 - Assess revenue distributions, downside risk, and robustness
-

12. Outputs of Interest

- Expected daily revenue
 - Revenue variance and downside risk
 - Probability of under-delivery
 - Sensitivity to penalty q_u
 - Impact of competing renewable penetration
-

Summary

This formulation yields a **two-layer stochastic simulation model**:

- **Aleatory uncertainty**: weather, demand, price noise
- **Epistemic uncertainty**: competitor bidding behavior

It is directly extensible to:

- Risk-averse objectives (e.g. CVaR)
- Multi-day investment analysis
- Strategic bidding or learning models

If you'd like, I can also:

- Translate this into a **mathematical program**
- Provide **pseudo-code** for Monte-Carlo simulation
- Extend it to **robust or distributionally robust optimization**



The means and variances of the market demand D_t and the wind generation G_t are also constants. Please refine your specification.

Below is a **refined and clarified specification** of the simulation model, explicitly distinguishing **constants**, **stochastic variables**, and **deep uncertainties**, and incorporating your clarification that the **means and variances of market demand and wind generation are also constants**.

1. Time and Market Structure

- **Time index:**

$$t = 1, \dots, 24$$

- **Market:** Day-ahead, hourly, uniform-price spot market cleared via **merit order**.
- **Producers:**
 - Conventional: $i \in \mathcal{C} = \{1, 2, 3\}$
 - Solar: s
 - Wind: w (decision maker)

2. Constant Parameters (Known and Fixed)

These parameters are **assumed known and fixed** in the simulation.

Demand

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$$

- μ_D : mean hourly demand
- σ_D^2 : variance of hourly demand

Wind Generation

$$G_t \sim \mathcal{N}(\mu_G, \sigma_G^2)$$

- μ_G : mean wind generation
- σ_G^2 : variance of wind generation

Solar Production Profile

$$b_{st} = \max\left(0, a + b \cos\left(\frac{2\pi t}{24}\right)\right)$$

with constants a, b .

Conventional Producer Quantities

$$b_{it} = \bar{b}_i, \quad i \in \mathcal{C}$$

Imbalance Penalty

- $q_u > 0$: under-delivery penalty (€/MWh)

3. Deep Uncertainties (Epistemic)

The following parameters are **not known with confidence** and define **scenarios** in the simulation:

Conventional Producer Bidding

$$p_{it} \sim \mathcal{N}(\mu_{pi}, \sigma_{pi}^2), \quad i \in \mathcal{C}$$

- μ_{pi}, σ_{pi}^2 : unknown bidding behavior parameters

Solar Producer Bidding

$$p_{st} \sim \mathcal{N}(\mu_{ps}, \sigma_{ps}^2)$$

- μ_{ps}, σ_{ps}^2 : unknown solar bidding behavior

These parameters vary across **long-term scenarios**, but remain fixed within a scenario.

4. Stochastic Variables (Aleatory Uncertainty)

Conditioned on fixed parameters, the following variables are random and independently sampled for each hour:

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$$

$$G_t \sim \mathcal{N}(\mu_G, \sigma_G^2)$$

$$p_{it} \sim \mathcal{N}(\mu_{pi}, \sigma_{pi}^2)$$

$$p_{st} \sim \mathcal{N}(\mu_{ps}, \sigma_{ps}^2)$$

5. Decision Variables (Wind Producer)

Chosen **day-ahead**:

$$(b_{wt}, p_{wt}), \quad t = 1, \dots, 24$$

where:

- $b_{wt} \geq 0$: bid quantity (MWh)

- $p_{wt} \geq 0$: bid price (€/MWh)

6. State Variables

At each hour t , the realized state is:

$$S_t = (D_t, G_t, \{p_{it}\}_{i \in \mathcal{C}}, p_{st})$$

No intertemporal state dependence is assumed.

7. Market Clearing (Transition Function)

Given bids:

$$\mathcal{B}_t = \{(b_{it}, p_{it})\}_{i \in \mathcal{C}} \cup (b_{st}, p_{st}) \cup (b_{wt}, p_{wt})$$

1. Sort bids by ascending price.
2. Accept bids until cumulative supply meets D_t .
3. Market-clearing price:

$$c_t = \max\{p : (b, p) \text{ accepted at } t\}$$

8. Wind Dispatch and Imbalance

Accepted Quantity

$$x_{wt} = \begin{cases} b_{wt}, & p_{wt} \leq c_t \\ 0, & p_{wt} > c_t \end{cases}$$

Under-Delivery

$$u_t = \max(0, x_{wt} - G_t)$$

9. Revenue Model

Hourly Revenue

$$R_t = c_t x_{wt} - q_u u_t$$

Daily Revenue

$$R = \sum_{t=1}^{24} R_t$$

10. Objective Function

The wind producer seeks to maximize **expected daily revenue**:

$$\max_{\{b_{wt}, p_{wt}\}_{t=1}^{24}} \mathbb{E} \left[\sum_{t=1}^{24} (c_t x_{wt} - q_u \max(0, x_{wt} - G_t)) \right]$$

Expectation is taken over:

$$(D_t, G_t, p_{it}, p_{st})$$

11. Simulation Structure (Final)

Inner Loop (Short-Term Uncertainty)

- Sample D_t, G_t, p_{it}, p_{st}
- Clear market
- Compute revenues

Outer Loop (Long-Term Scenarios)

- Vary deep uncertainty parameters
($\mu_{pi}, \sigma_{pi}^2, \mu_{ps}, \sigma_{ps}^2$)
- Evaluate robustness and downside risk

12. Key Refinement Summary

Component	Classification
μ_D, σ_D^2	Constants
μ_G, σ_G^2	Constants
D_t, G_t	Aleatory stochastic variables
Competitor price parameters	Deep uncertainty
Wind bids (b_{wt}, p_{wt})	Decision variables

If you want, I can now:

- Reformulate this as a **two-stage stochastic program**
- Add **correlation structures** (e.g. wind-solar, demand-price)
- Introduce **risk measures** (CVaR, chance constraints)



Various stakeholders approach this problem from different perspectives. Please refine the model specification from the perspectives of (1) the wind-power producer; and (2) the system regulator. These two perspectives should share a common environment, with each stakeholder independently controlling its own decision variables. In the specification, specify the common environment, including the global parameters and variables. For each perspective, also specify the corresponding model and its key components, including the decision variables, transition functions, and objective functions. If there are multiple objectives, provide multiple objective functions instead of a weighted objective function.

Below is a **refined multi-stakeholder specification** that cleanly separates the **common environment** from the **individual decision problems** of

- (1) the **wind-power producer**, and
- (2) the **system regulator**.

The two stakeholders **share the same stochastic market environment**, but **control different decision variables** and pursue **distinct objectives**.

I. Common Environment (Shared by All Stakeholders)

The environment defines the physical system, market rules, uncertainties, and exogenous parameters. All agents observe the same realizations but act independently.

I.1 Time and Market Structure

- Time horizon:

$$t = 1, \dots, 24$$

- Market:
 - Day-ahead, hourly spot market
 - Uniform pricing
 - Merit-order clearing
 - No strategic coordination among producers

I.2 Global Constant Parameters

These parameters are **known and fixed**.

Demand Distribution

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$$

- μ_D : mean hourly demand
- σ_D^2 : variance of hourly demand

Wind Generation Distribution

$$G_t \sim \mathcal{N}(\mu_G, \sigma_G^2)$$

- μ_G : mean wind generation
- σ_G^2 : variance of wind generation

Solar Production Profile

$$b_{st} = \max\left(0, a + b \cos\left(\frac{2\pi t}{24}\right)\right)$$

with constants a, b .

Conventional Production Capacities

$$b_{it} = \bar{b}_i, \quad i \in \mathcal{C} = \{1, 2, 3\}$$

I.3 Deep Uncertainties (Epistemic)

These parameters are **unknown to all agents** and define **long-term scenarios**:

- Conventional bidding behavior:

$$p_{it} \sim \mathcal{N}(\mu_{pi}, \sigma_{pi}^2)$$

- Solar bidding behavior:

$$p_{st} \sim \mathcal{N}(\mu_{ps}, \sigma_{ps}^2)$$

The parameters

$$(\mu_{pi}, \sigma_{pi}^2, \mu_{ps}, \sigma_{ps}^2)$$

are fixed within a scenario but vary across scenarios.

I.4 Stochastic Variables (Aleatory)

Conditioned on parameters, hourly random variables:

$$\{D_t, G_t, p_{it}, p_{st}\}$$

Independent across hours.

I.5 Market Clearing (Shared Transition Rule)

At each hour t :

1. Collect bids:

$$\mathcal{B}_t = \{(b_{it}, p_{it})\}_{i \in \mathcal{C}} \cup (b_{st}, p_{st}) \cup (b_{wt}, p_{wt})$$

2. Sort bids by ascending price.
3. Accept bids until total accepted supply meets D_t .
4. Market-clearing price:

$$c_t = \max\{p : (b, p) \text{ accepted}\}$$

II. Perspective 1: Wind-Power Producer

The wind producer is a **profit-seeking agent** operating under uncertainty.

II.1 State Variables (Observed)

At hour t :

$$S_t^{(w)} = (D_t, c_t, G_t)$$

II.2 Decision Variables

Chosen **day-ahead**:

$$(b_{wt}, p_{wt}), \quad t = 1, \dots, 24$$

- $b_{wt} \geq 0$: bid quantity
- $p_{wt} \geq 0$: bid price

II.3 Transition Functions

Dispatch Decision

$$x_{wt} = \begin{cases} b_{wt}, & p_{wt} \leq c_t \\ 0, & p_{wt} > c_t \end{cases}$$

Imbalance

$$u_t = \max(0, x_{wt} - G_t)$$

II.4 Objective Functions

The wind producer may pursue **multiple objectives**.

Objective 1: Maximize Expected Profit

$$\max \mathbb{E} \left[\sum_{t=1}^{24} (c_t x_{wt} - q_u u_t) \right]$$

Objective 2: Minimize Downside Risk (Reliability Exposure)

$$\min \mathbb{E} \left[\sum_{t=1}^{24} u_t \right]$$

Objective 3: Minimize Revenue Variability

$$\min \text{Var} \left(\sum_{t=1}^{24} c_t x_{wt} \right)$$

(Handled via Pareto analysis, not scalarization.)

III. Perspective 2: System Regulator

The regulator **does not bid energy**, but **sets market rules** to ensure system reliability and economic efficiency.

III.1 State Variables (Observed)

$$S_t^{(r)} = (D_t, \{x_{jt}\}, G_t, u_t, c_t)$$

where j indexes all producers.

III.2 Decision Variables

The regulator controls **policy parameters**, fixed for the day:

- Imbalance penalty:

$$q_u \geq 0$$

- (Optional extensions)
 - Price caps c^{\max}
 - Renewable priority rules
 - Reserve requirements

III.3 Transition Functions

The regulator **does not alter clearing directly**, but influences outcomes through q_u , which affects:

- Wind bidding behavior
- Under-delivery incentives
- Real-time system balance

III.4 Objective Functions

The regulator has **multiple, non-commensurate objectives**.

Objective 1: Minimize Expected Supply Shortfall

$$\min \mathbb{E} \left[\sum_{t=1}^{24} \max(0, D_t - \sum_j x_{jt}) \right]$$

Objective 2: Minimize Total System Imbalance

$$\min \mathbb{E} \left[\sum_{t=1}^{24} \sum_j |x_{jt} - G_{jt}| \right]$$

Objective 3: Minimize Consumer Cost

$$\min \mathbb{E} \left[\sum_{t=1}^{24} c_t D_t \right]$$

Objective 4: Support Renewable Integration

$$\max \mathbb{E} \left[\sum_{t=1}^{24} x_{wt} \right]$$

IV. Interaction Structure Summary

Layer	Description
Environment	Demand, generation, bidding distributions, clearing rules
Wind Producer	Chooses (b_{wt}, p_{wt}) to trade profit vs risk
Regulator	Chooses q_u to trade efficiency vs reliability
Coupling	Wind decisions respond to q_u ; system outcomes feed regulator objectives

V. Interpretation

- This is a **multi-agent stochastic system** with **shared uncertainty**
- The wind producer solves a **stochastic control / bidding problem**
- The regulator solves a **policy design problem**
- No agent internalizes the other's objective → **strategic misalignment**
- The framework supports:
 - Scenario discovery
 - Robust policy design
 - Market-design stress testing

If you want next steps, I can:

- Reformulate this as a **bi-level stochastic game**
- Add **learning or adaptive bidding**
- Introduce **distributionally robust objectives**



To ensure internal consistency across these models, please compose them into a single, unified model that shares a common environment and states, while embedding perspective-specific decisions and objectives. The composed model should: 1. embed each perspective's decision variables and objectives; 2. enable modular evaluation of each perspective; 3. for each perspective, treat other perspectives' decisions as exogenous inputs; and 4. incorporate all other components from every perspective.

Below is a **single, unified composed model** that ensures **internal consistency**, while allowing **modular, perspective-specific evaluation** for the **wind-power producer** and the **system regulator**. The model is structured as a **shared stochastic environment** with **multiple decision-making modules**, each treating the others' decisions as **exogenous inputs**.

Unified Composed Model

1. Global Structure

The composed model is defined as a tuple:

$$\mathcal{M} = \left(\mathcal{T}, \mathcal{E}, \{\mathcal{A}^k\}_{k \in \{w, r\}}, \mathcal{P}, \{\mathcal{O}^k\}_{k \in \{w, r\}} \right)$$

where:

- \mathcal{T} : time structure
- \mathcal{E} : shared environment and state space
- \mathcal{A}^k : decision module of stakeholder k
- \mathcal{P} : transition and market-clearing dynamics
- \mathcal{O}^k : objective functions of stakeholder k

2. Time Structure

$$\mathcal{T} = \{1, \dots, 24\}$$

All decisions are **day-ahead** and all uncertainty resolves **hourly**.

3. Shared Environment and State Space

3.1 Global Parameters (Constants)

$$\Theta = \{\mu_D, \sigma_D^2, \mu_G, \sigma_G^2, a, b, \{\bar{b}_i\}_{i \in \mathcal{C}}\}$$

These parameters are **fixed and known** to all perspectives.

3.2 Deep Uncertainty Parameters (Scenario Inputs)

$$\Xi = \{\mu_{pi}, \sigma_{pi}^2, \mu_{ps}, \sigma_{ps}^2\}$$

These parameters are:

- Fixed within a scenario
- Exogenous to all decision makers
- Used for long-term stress testing

3.3 Stochastic Variables

For each hour t :

$$\omega_t = (D_t, G_t, \{p_{it}\}_{i \in \mathcal{C}}, p_{st})$$

with:

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$$

$$G_t \sim \mathcal{N}(\mu_G, \sigma_G^2)$$

$$p_{it} \sim \mathcal{N}(\mu_{pi}, \sigma_{pi}^2)$$

$$p_{st} \sim \mathcal{N}(\mu_{ps}, \sigma_{ps}^2)$$

3.4 Global State

At hour t , the shared system state is:

$$S_t = (\omega_t, c_t, \{x_{jt}\}_{j \in \mathcal{J}}, \{u_{jt}\}_{j \in \mathcal{J}})$$

where:

- c_t : market-clearing price
- x_{jt} : dispatched quantity
- u_{jt} : imbalance (under-delivery)
- $\mathcal{J} = \mathcal{C} \cup \{s, w\}$

4. Decision Modules (Perspective-Specific)

Each perspective controls its own decisions while treating others' decisions as **exogenous parameters**.

4.1 Wind-Power Producer Module ($k = w$)

Decision Variables

$$A^w = \{(b_{wt}, p_{wt})\}_{t=1}^{24}$$

Exogenous Inputs

$$A^r = \{q_u\}$$

(the regulator's penalty choice)

4.2 System Regulator Module ($k = r$)

Decision Variables

$$A^r = \{q_u\}$$

Exogenous Inputs

$$A^w = \{(b_{wt}, p_{wt})\}_{t=1}^{24}$$

(the wind producer's bidding strategy)

5. Shared Transition and Market-Clearing Dynamics

5.1 Merit-Order Clearing

For each hour t :

1. Construct bid set:

$$\mathcal{B}_t = \{(\bar{b}_i, p_{it})\}_{i \in \mathcal{C}} \cup (b_{st}, p_{st}) \cup (b_{wt}, p_{wt})$$

2. Sort bids by ascending price.

3. Accept bids until:

$$\sum_j x_{jt} \geq D_t$$

4. Market-clearing price:

$$c_t = \max\{p : (b, p) \text{ accepted}\}$$

5.2 Dispatch and Imbalance

For each producer j :

$$x_{jt} = \begin{cases} b_{jt}, & p_{jt} \leq c_t \\ 0, & \text{otherwise} \end{cases}$$

Wind under-delivery:

$$u_{wt} = \max(0, x_{wt} - G_t)$$

6. Perspective-Specific Objective Functions

The environment produces **common trajectories**, which are evaluated **separately** by each stakeholder.

6.1 Wind-Power Producer Objectives (\mathcal{O}^w)

Objective 1: Maximize Expected Profit

$$\max_{A^w} \mathbb{E} \left[\sum_{t=1}^{24} (c_t x_{wt} - q_u u_{wt}) \right]$$

Objective 2: Minimize Expected Imbalance

$$\min_{A^w} \mathbb{E} \left[\sum_{t=1}^{24} u_{wt} \right]$$

Objective 3: Minimize Revenue Variance

$$\min_{A^w} \text{Var} \left(\sum_{t=1}^{24} c_t x_{wt} \right)$$

6.2 System Regulator Objectives (\mathcal{O}^r)

Objective 1: Minimize Expected System Shortfall

$$\min_{A^r} \mathbb{E} \left[\sum_{t=1}^{24} \max \left(0, D_t - \sum_j x_{jt} \right) \right]$$

Objective 2: Minimize Total Imbalance

$$\min_{A^r} \mathbb{E} \left[\sum_{t=1}^{24} \sum_j |x_{jt} - G_{jt}| \right]$$

Objective 3: Minimize Consumer Payments

$$\min_{A^r} \mathbb{E} \left[\sum_{t=1}^{24} c_t D_t \right]$$

Objective 4: Maximize Wind Utilization

$$\max_{A^r} \mathbb{E} \left[\sum_{t=1}^{24} x_{wt} \right]$$

7. Modularity and Evaluation Logic

- **Single simulation run** produces:

$$\{S_t\}_{t=1}^{24}$$

- Each perspective:
 - Takes other agents' actions as **fixed inputs**
 - Evaluates its **own objectives only**
 - Does not internalize other objectives

This allows:

- Independent policy evaluation
- Counterfactual analysis
- Scenario discovery
- Robustness testing

8. Interpretation

This unified model is:

- **Internally consistent** (single environment, single state)
- **Perspective-modular** (separable objectives)
- **Non-cooperative by construction**
- Suitable for:
 - Bi-level or equilibrium extensions
 - Robust decision-making under deep uncertainty
 - Market design stress testing

If you'd like, I can next:

- Reformulate this as a **stochastic game**
- Provide **simulation pseudocode**
- Extend it to **adaptive or learning agents**



Now, provide a modular Python implementation of the unified model, using classes to represent different perspectives. At the end of the program, define a function as the interface. This function should take as inputs a dictionary of decision variable values, a dictionary of uncertain parameter values, and a dictionary of constant parameter values; simulate the model dynamics; and return the value of each objective function. This function does not need to be executed.

Below is a **modular Python implementation** of the **unified composed model**.

It is written to be **internally consistent**, **perspective-modular**, and **simulation-ready**, while remaining **agnostic about optimization**.

Key design choices:

- A **shared Environment** class encapsulates uncertainty, market clearing, and state evolution.
- Each **Perspective** (Wind Producer, Regulator) is a class with its own objectives.
- Other perspectives' decisions are treated as **exogenous inputs**.
- A **single interface function** runs the simulation and evaluates all objectives.

⚠ This code is a *specification-level implementation*: it is not intended to be executed as-is, but it is complete and logically consistent.

1. Core Imports

```
python

import numpy as np
```

2. Shared Environment

```
python

class MarketEnvironment:
    """
    Shared stochastic environment:
    - samples uncertainty
    - clears the market
    - produces system-wide trajectories
    """

    def __init__(self, constants, uncertainties, decisions):
        self.constants = constants
        self.uncertainties = uncertainties
        self.decisions = decisions

        self.T = 24
        self.conventional_ids = [0, 1, 2]

    def sample_uncertainty(self):
        """Sample all stochastic variables for one day."""
```

```

mu_D, sigma_D = self.constants["mu_D"], self.constants["sigma_D"]
mu_G, sigma_G = self.constants["mu_G"], self.constants["sigma_G"]

D = np.random.normal(mu_D, sigma_D, self.T)
G = np.random.normal(mu_G, sigma_G, self.T)

p_conv = {}
for i in self.conventional_ids:
    mu_pi, sigma_pi = self.uncertainties["mu_pi"][i], self.uncertainties["sigma_pi"][i]
    p_conv[i] = np.random.normal(mu_pi, sigma_pi, self.T)

p_solar = np.random.normal(
    self.uncertainties["mu_ps"],
    self.uncertainties["sigma_ps"],
    self.T
)

return D, G, p_conv, p_solar

def solar_quantity(self, t):
    a, b = self.constants["a"], self.constants["b"]
    return max(0.0, a + b * np.cos(2 * np.pi * (t + 1) / 24))

def clear_market(self, t, D_t, p_conv_t, p_solar_t, wind_bid):
    """
    Merit-order market clearing for one hour.
    """
    bids = []

    # Conventional producers
    for i in self.conventional_ids:
        bids.append((
            self.constants["b_conv"][i],
            p_conv_t[i],
            f"conv_{i}"
        ))

    # Solar producer
    bids.append((
        self.solar_quantity(t),
        p_solar_t,
        "solar"
    ))

    # Wind producer
    b_w, p_w = wind_bid
    bids.append((b_w, p_w, "wind"))

    # Sort bids by price
    bids.sort(key=lambda x: x[1])

    accepted = {}
    supplied = 0.0
    clearing_price = 0.0

    for qty, price, name in bids:
        if supplied >= D_t:
            break
        accepted[name] = qty
        supplied += qty
        clearing_price = price

    return accepted, clearing_price

```

3. Wind-Power Producer Perspective

python

```

class WindProducer:
    """

```

```

Wind producer perspective:
evaluates profit and risk objectives
"""

def __init__(self, decisions, constants):
    self.bids = decisions["wind_bids"]
    self.q_u = decisions["q_u"]
    self.constants = constants

def evaluate(self, trajectory):
    profits = []
    imbalances = []

    for t in range(24):
        c_t = trajectory["prices"][t]
        x_w = trajectory["dispatch"]["wind"][t]
        G_t = trajectory["wind_generation"][t]

        under_delivery = max(0.0, x_w - G_t)
        profit = c_t * x_w - self.q_u * under_delivery

        profits.append(profit)
        imbalances.append(under_delivery)

    return {
        "expected_profit": np.sum(profits),
        "total_imbalance": np.sum(imbalances),
        "revenue_variance": np.var(profits),
    }

```

4. System Regulator Perspective

python

```

class SystemRegulator:
    """
    Regulator perspective:
    evaluates system-wide reliability and cost objectives
    """

    def __init__(self):
        pass

    def evaluate(self, trajectory):
        shortages = []
        total_imbalance = []
        consumer_costs = []
        wind_utilization = []

        for t in range(24):
            D_t = trajectory["demand"][t]
            total_supply = sum(
                trajectory["dispatch"][agent][t]
                for agent in trajectory["dispatch"]
            )

            shortage = max(0.0, D_t - total_supply)
            shortages.append(shortage)

            imbalance_t = sum(
                abs(trajectory["dispatch"][agent][t] -
                    trajectory["realized_generation"].get(agent, 0.0))
                for agent in trajectory["dispatch"]
            )
            total_imbalance.append(imbalance_t)

            consumer_costs.append(trajectory["prices"][t] * D_t)
            wind_utilization.append(trajectory["dispatch"]["wind"][t])

        return {

```

```

        "expected_shortfall": np.sum(shortages),
        "total_system_imbalance": np.sum(total_imbalance),
        "total_consumer_payment": np.sum(consumer_costs),
        "total_wind_dispatch": np.sum(wind_utilization),
    }

```

5. Unified Simulation Interface Function

python

```

def simulate_unified_model(decisions, uncertainties, constants):
    """
    Unified simulation interface.

    Parameters
    -----
    decisions : dict
        - "wind_bids": list of (b_wt, p_wt) for t=1..24
        - "q_u": imbalance penalty

    uncertainties : dict
        - "mu_pi": list of means for conventional bids
        - "sigma_pi": list of std devs for conventional bids
        - "mu_ps": mean solar bid price
        - "sigma_ps": std dev solar bid price

    constants : dict
        - demand, wind, solar, and capacity parameters

    Returns
    -----
    dict
        Objective values for each perspective
    """

    env = MarketEnvironment(constants, uncertainties, decisions)

    D, G, p_conv, p_solar = env.sample_uncertainty()

    trajectory = {
        "demand": D,
        "wind_generation": G,
        "prices": [],
        "dispatch": {"wind": [], "solar": [], "conv_0": [], "conv_1": [], "conv_2": []},
        "realized_generation": {"wind": G},
    }

    for t in range(24):
        accepted, price = env.clear_market(
            t,
            D[t],
            {i: p_conv[i][t] for i in env.conventional_ids},
            p_solar[t],
            decisions["wind_bids"][t],
        )

        trajectory["prices"].append(price)

        for agent in trajectory["dispatch"]:
            trajectory["dispatch"][agent].append(accepted.get(agent, 0.0))

    wind_agent = WindProducer(decisions, constants)
    regulator = SystemRegulator()

    wind_objectives = wind_agent.evaluate(trajectory)
    regulator_objectives = regulator.evaluate(trajectory)

    return {
        "wind_producer_objectives": wind_objectives,

```



```
    "regulator_objectives": regulator_objectives,  
}
```

6. What This Implementation Enables

- ☒ Single shared stochastic environment
- ☒ Modular evaluation of each perspective
- ☒ Exogenous treatment of other agents' decisions
- ☒ Easy extension to:
 - Monte Carlo loops
 - Robust or distributionally robust analysis
 - Learning or adaptive strategies

If you want next, I can:

- Add **Monte Carlo wrappers**
- Convert this into a **bi-level optimization template**
- Extend it with **correlated uncertainties**



Please refine your implementation so that any bid with a price less than or equal to the clearing price is accepted, even if the total market demand has already been met.

Good catch — you're right.

The earlier implementation implicitly enforced **partial acceptance** by stopping once demand was met, which is **not consistent with the stated uniform-price, merit-order rule**.

Below is a **refined implementation** that ensures:

Any bid with price \leq clearing price is fully accepted,
even if aggregate accepted supply exceeds demand.

I'll first explain the logic change, then give the **corrected code**. Everything else in the model remains internally consistent.

Conceptual Refinement (What Changes)

The correct clearing logic is:

1. **Sort bids by price**
2. **Find the clearing price:**
 - The *lowest price* such that cumulative supply **meets or exceeds demand**
3. **Accept all bids with price \leq clearing price**
 - No partial acceptance
 - Oversupply is allowed
 - All accepted bids are paid the same clearing price

This mirrors standard **uniform-price auction** logic.

Refined Market-Clearing Implementation

✓ Corrected `clear_market` Method

```
python

def clear_market(self, t, D_t, p_conv_t, p_solar_t, wind_bid):
    """
    Merit-order market clearing with uniform pricing:
    all bids with price <= clearing price are accepted.
    """

    bids = []

    # Conventional producers
    for i in self.conventional_ids:
```

```

        bids.append((
            self.constants["b_conv"][i],
            p_conv_t[i],
            f"conv_{i}"
        ))

# Solar producer
bids.append((
    self.solar_quantity(t),
    p_solar_t,
    "solar"
))

# Wind producer
b_w, p_w = wind_bid
bids.append((b_w, p_w, "wind"))

# Sort bids by increasing price
bids.sort(key=lambda x: x[1])

# Step 1: determine clearing price
cumulative_supply = 0.0
clearing_price = None

for qty, price, _ in bids:
    cumulative_supply += qty
    if cumulative_supply >= D_t:
        clearing_price = price
        break

# Step 2: accept all bids with price <= clearing price
accepted = {}
for qty, price, name in bids:
    if price <= clearing_price:
        accepted[name] = qty
    else:
        accepted[name] = 0.0

return accepted, clearing_price

```

Why This Fix Matters

This refinement ensures:

- ✓ **Logical consistency with your model specification**
- ✓ Wind may be dispatched even when marginal
- ✓ Oversupply is possible (important for imbalance analysis)
- ✓ Clearing price is **well-defined and unique**
- ✓ Regulator objectives (imbalance, consumer cost) are now meaningful

What Remains Unchanged (and Correct)

- Wind under-delivery penalties are still computed **ex post**
- The regulator does **not** ration supply
- Wind, solar, and conventional producers are treated symmetrically
- Perspective modularity is preserved

If you'd like, I can now:

- Add **partial acceptance as an optional market design variant**
- Make clearing price **robust to extreme demand realizations**
- Add **price caps or scarcity pricing rules**