In this problem, a renewable energy producer participates in a day-ahead merit-order electricity market competing against three other conventional producers. The producer submits hourly bids that specify a delivery quantity of electricity and an offer price for each hour of the next day. If a bid is accepted, the producer is obligated to deliver the specified quantity of electricity at the offer price. Otherwise, the producer earns zero profit for that hour. However, renewable generation is deeply uncertain, as it depends on weather conditions that cannot be predicted. Meanwhile, the hourly clearing price depends on the bids of all producers and is unknown in advance. The renewable energy producer's goal is to maximize its expected profit by strategically selecting the quantity of electricity b_t and offer price p_t to bid for each hour of the next day, accounting for the uncertainty in both renewable generation and clearing prices.

Please formalize this problem description as a model. Please also specify key components, such as state variables, stochastic variables, constant parameters, uncertainties and transition functions.

Excellent — this is a rich stochastic optimization and decision-making problem in an energy market context. Let's carefully formalize it as a **stochastic bidding model** for a renewable producer in a day-ahead merit-order market.

---

1. Model Overview

The renewable producer decides **hourly bids** (b_t, p_t) for $t = 1, \dots, T$, where each bid specifies:

- b_t: quantity of energy offered (MWh), and
- p_t: offer price ($/MWh).

The producer's **profit** depends on whether the bid is accepted in the market (which depends on the **market clearing price**) and on whether the renewable generation is sufficient to meet the committed delivery.

Uncertainties arise from:

- **Market clearing price** π_t (random, depends on other producers' bids),
- **Renewable generation** g_t (random, depends on weather).

The producer aims to **maximize expected profit** over the planning horizon (e.g., 24 hours).

---

2. Sets and Indices

Symbol	Meaning
$t \in \mathcal{T} = \{1, \dots, T\}$	Time periods (e.g., hours in a day)

---

3. Decision Variables (controlled by the renewable producer)

Symbol	Description
$b_t \geq 0$	Quantity of electricity offered (MWh)
$p_t \geq 0$	Offer price ($/MWh)

The decision vector for all hours is $\mathbf{x} = \{(b_t, p_t)\}_{t \in \mathcal{T}}$.

4. Stochastic Variables (random outcomes)

Symbol	Description	Distribution/Source
g_t	Available renewable generation (MWh)	depends on weather
π_t	Market clearing price ($/MWh)	depends on other bids / demand

Both are random variables known only through **probabilistic forecasts** at the time of bidding.

5. Constants and Parameters

Symbol	Description
c_t	Penalty cost per MWh of shortfall (if $b_t > g_t$)
$\bar{b}_t$	Maximum bid (physical or regulatory limit)
Ω	Scenario set representing possible realizations of (π_t, g_t)
$\mathbb{P}(\omega)$	Probability of scenario $\omega \in \Omega$

6. Market Acceptance Rule

In a merit-order market:

- The bid is **accepted** if $p_t \leq \pi_t$.
- Otherwise, it is rejected.

Define the binary acceptance variable:

$$a_t(\omega) = \begin{cases} 1, & \text{if } p_t \leq \pi_t(\omega), \\ 0, & \text{otherwise.} \end{cases}$$

7. Payoff Function (per hour, per scenario)

If the bid is accepted:

- The producer must deliver b_t MWh.
- If actual generation $g_t \geq b_t$, the delivery is feasible.

- If $g_t < b_t$, the producer pays a penalty for the shortfall $(b_t - g_t)_+$.

Hence, **profit in hour t under scenario ω** is:

$$\pi_t^{\text{profit}}(\omega) = a_t(\omega)\left[p_t b_t - c_t(b_t - g_t(\omega))_+\right].$$

If the bid is not accepted ($a_t = 0$), profit = 0.

---

8. Objective Function

The renewable producer maximizes expected total profit:

$$\max_{\{b_t,p_t\}_{t\in\mathcal{T}}} \mathbb{E}_{\pi,g}\left[\sum_{t=1}^{T} a_t(\pi_t, p_t)\left(p_t b_t - c_t(b_t - g_t)_+\right)\right].$$

In scenario form:

$$\max_{\{b_t,p_t\}} \sum_{\omega\in\Omega} \mathbb{P}(\omega) \sum_{t=1}^{T} 1_{\{p_t\le\pi_t(\omega)\}}\left[p_t b_t - c_t(b_t - g_t(\omega))_+\right].$$

subject to:

$$0 \le b_t \le \bar{b}_t, \quad p_t \ge 0, \quad \forall t \in \mathcal{T}.$$

---

9. State Variables and Transition Dynamics

Because bids are set day-ahead for all hours simultaneously, the system is **static within a day** (no feedback between hours), unless the model includes energy storage or ramping constraints.

However, if we extend to a multi-day or adaptive bidding problem, we define:

Symbol	Meaning
s_t	System state at hour t (e.g., remaining energy, forecast updates, financial position)
$s_{t+1} = f(s_t, b_t, p_t, g_t, \pi_t)$	State transition function

Example (if including forecast updating):

$$s_{t+1} = \text{UpdateForecast}(s_t, \text{new info at } t+1).$$

In the one-day setting, this is often omitted (static optimization).

---

10. Compact Model Representation

$$\begin{aligned}
\max_{\{b_t,p_t\}} \quad & \mathbb{E}_{\pi,g}\left[\sum_{t=1}^{T} 1_{\{p_t\le\pi_t\}}\left(p_t b_t - c_t(b_t - g_t)_+\right)\right] \\
\text{s.t.} \quad & 0 \le b_t \le \bar{b}_t, \quad p_t \ge 0, \quad \forall t.
\end{aligned}$$

---

11. Interpretation

- The **expected profit** depends jointly on **price risk** (whether the bid is accepted) and **quantity risk** (whether renewable output meets commitments).
- The model is **nonlinear and stochastic**, due to the indicator function and expectation over random prices and generation.
- Extensions can include risk aversion (e.g., CVaR objective), multi-stage dynamics, or co-optimization with storage.

Would you like me to extend this into a **two-stage stochastic programming formulation** (with first-stage bidding decisions and second-stage realization of uncertainty)? That would make the structure more explicit for numerical solution.

First, the day-ahead market is cleared once per day for 24 hourly intervals of the following day, so you can assume the problem only considers a 24-hour period. Second, penalties would apply if the renewable producer under-delivers or over-delivers relative to its bid. Please refine the specification.

Excellent — your clarification (single-day horizon with 24 hourly markets and symmetric penalties for both under- and over-delivery) makes the structure sharper.
Let's now restate and formalize the model **precisely** for a one-day, 24-hour day-ahead market, capturing both sources of uncertainty and symmetric imbalance penalties.

🧭 Problem Scope

- The renewable producer submits **24 hourly bids** (b_t, p_t), $t = 1, \dots, 24$, in the **day-ahead market**.
- The **day-ahead market** is cleared once per day, producing **hourly clearing prices** π_t for the next day.
- Actual **renewable generation** g_t is realized the next day and may differ from b_t.
- Deviations $g_t - b_t$ incur **imbalance penalties** (either for shortfall or surplus).

---

1. Sets and Indices

Symbol	Meaning
$t \in \mathcal{T} = \{1, 2, \dots, 24\}$	Hours in the next day
$\omega \in \Omega$	Scenario index for uncertain outcomes

---

2. Decision Variables (chosen day-ahead)

Symbol	Description
$b_t \geq 0$	Quantity offered to deliver at hour t (MWh)
$p_t \geq 0$	Offer price for that quantity ($/MWh)

These are **first-stage** decisions made before any uncertainty is realized.

---

3. Stochastic Variables (realized after market clearing and operation)

Symbol	Description
$\pi_t(\omega)$	Market clearing price ($/MWh) in scenario ω
$g_t(\omega)$	Actual renewable generation (MWh) in scenario ω

Both are random variables whose distributions or scenario sets are estimated from forecasts.

---

4. Market Acceptance Rule (clearing mechanism)

For each hour t and scenario ω:

$$a_t(\omega) = \begin{cases} 1, & \text{if } p_t \leq \pi_t(\omega) \quad \text{(bid accepted)}, \\ 0, & \text{otherwise}. \end{cases}$$

If the bid is not accepted, no energy is delivered and no payment occurs.

---

5. Imbalance Settlement and Penalties

When the bid is accepted ($a_t = 1$), the producer is obligated to deliver b_t MWh.
However, the realized generation $g_t(\omega)$ may differ. The **imbalance quantity** is:

$$\delta_t(\omega) = g_t(\omega) - b_t.$$

- If $\delta_t(\omega) < 0$: shortfall → under-delivery.
- If $\delta_t(\omega) > 0$: surplus → over-delivery.

Let:

- c_t^-: penalty rate ($/MWh) for **under-delivery** (when $g_t < b_t$);
- c_t^+: penalty rate ($/MWh) for **over-delivery** (when $g_t > b_t$).

The penalty function is:

$$\text{Penalty}_t(\omega) = c_t^-(b_t - g_t(\omega))_+ + c_t^+(g_t(\omega) - b_t)_+.$$

---

6. Profit Function (per hour, per scenario)

If the bid is accepted:

$$\Pi_t(\omega) = p_t b_t - \text{Penalty}_t(\omega) = p_t b_t - c_t^-(b_t - g_t(\omega))_+ - c_t^+(g_t(\omega) - b_t)_+.$$

If not accepted ($a_t = 0$), profit = 0.

So the realized profit is:

$$\Pi_t(\omega) = a_t(\omega)\left[p_t b_t - c_t^-(b_t - g_t(\omega))_+ - c_t^+(g_t(\omega) - b_t)_+\right].$$

7. Objective Function

The renewable producer maximizes **expected total profit** over 24 hours:

$$\max_{\{b_t, p_t\}_{t\in\mathcal{T}}} \mathbb{E}_{\pi,g}\left[\sum_{t=1}^{24} a_t(\pi_t, p_t)\left(p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+\right)\right].$$

Or equivalently, in scenario-based form:

$$\max_{\{b_t, p_t\}} \sum_{\omega\in\Omega} \mathbb{P}(\omega) \sum_{t=1}^{24} 1_{\{p_t \le \pi_t(\omega)\}}\left[p_t b_t - c_t^-(b_t - g_t(\omega))_+ - c_t^+(g_t(\omega) - b_t)_+\right].$$

subject to:

$$0 \le b_t \le \bar{b}_t, \quad p_t \ge 0, \quad \forall t \in \mathcal{T}.$$

8. Key Model Components (Structured Summary)

Type	Symbol	Description
Decision variables	b_t, p_t	Bid quantity and offer price
State variables	none (single-day static problem)	
Stochastic variables	g_t, π_t	Renewable generation and clearing price
Parameters	$c_t^-, c_t^+, \bar{b}_t$	Penalty rates and bid limits
Uncertainties	Weather $\rightarrow g_t$; Market competition $\rightarrow \pi_t$	
Transition functions	none (no inter-hour dependency)	
Indicator function	$a_t(\omega) = 1_{\{p_t \le \pi_t(\omega)\}}$	Determines acceptance
Payoff	$a_t(p_t, \pi_t)(p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+)$	Realized profit

9. Model Type

This formulation is a **single-stage stochastic optimization problem** with **recourse-like penalties** (since imbalance costs depend on realized generation).

It can also be seen as a **bilevel stochastic game** if one models the market clearing process explicitly.

---

10. Optional Extensions

You could add:

- **Risk aversion**: replace expectation by CVaR or weighted utility of profit;
- **Correlated uncertainties**: model joint distribution of (π_t, g_t);
- **Inter-hour constraints**: if ramping or storage applies, add $b_{t+1} - b_t \leq R$;
- **Chance constraints**: ensure $\Pr(g_t \geq b_t) \geq 1 - \epsilon$ to limit shortfall risk.

Would you like me to now **express this formally as a two-stage stochastic program** (with explicit first-stage bids and second-stage imbalance realization), suitable for solving numerically with scenario decomposition (e.g., via extensive form or SAA)?

> There are still several areas where the model specification can be improved: 1. The renewable generation is deeply uncertain, so its probability distribution is unknown; 2. The market clearing price can be calculated based on the total energy demand and the bids of all energy producers for each hour.

Excellent — this is a key refinement.
You're pointing out two structural features that make this **a realistic, but nontrivial, stochastic–robust bidding model**:

1. **Renewable generation uncertainty is *deep* (distributionally unknown).**
   → This calls for a **distributionally robust** or **worst-case** treatment, rather than an expectation over a known probability distribution.
2. **Market clearing prices** are **endogenous** — determined by **merit-order clearing** given all producers' bids and system demand.
   → So π_t isn't an exogenous random variable, but a **deterministic function** of all producers' offers and total demand.

Let's now integrate these two aspects and refine the formal specification accordingly.

🧩 1. Model Context and Structure

We consider a **single-day (24-hour)** day-ahead market involving:

- One **renewable producer** r (the decision-maker),
- Three **conventional competitors** $j \in \mathcal{J} = \{1, 2, 3\}$,
- A **system operator** clearing the market for each hour $t = 1, \ldots, 24$,

- A **fixed demand** D_t (deterministic or forecasted).

The renewable producer selects bids (b_t, p_t), anticipating:

- uncertain renewable generation g_t (deep uncertainty);
- endogenous market clearing price π_t determined by all offers.

🧮 2. Sets and Indices

Symbol	Meaning
$t \in \mathcal{T} = \{1, \ldots, 24\}$	Hours of next day
$j \in \mathcal{J}$	Competing producers
r	Renewable producer (decision-maker)

⚙️ 3. Decision Variables

Symbol	Description
$b_t \geq 0$	Quantity offered by renewable producer (MWh)
$p_t \geq 0$	Offer price ($/MWh)

All decisions are **day-ahead**, before market clearing and before generation is realized.

🌤️ 4. Uncertain Quantities

4.1 Renewable generation (deep uncertainty)

- $g_t \in \mathcal{G}_t \subseteq [0, \bar{g}_t]$: actual renewable generation.
- **No probability distribution** is assumed.
- Instead, uncertainty is represented by an **ambiguity set** or **bounded set** $\mathcal{G}_t$ derived from forecasts, e.g.:

$$\mathcal{G}_t = [\underline{g}_t, \overline{g}_t].$$

- The producer adopts a **robust or worst-case** approach, protecting against unfavorable realizations of g_t.

4.2 Market clearing price (endogenous)

Let:

- (b_t^j, p_t^j): bids of competitors $j \in \mathcal{J}$,
- D_t: total system demand.

For each hour t, the **market operator** clears the market by solving:

$$\begin{aligned} \min_{\pi_t, q_t^j, q_t^r} \quad & \pi_t \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}} q_t^j + q_t^r = D_t, \\ & 0 \leq q_t^j \leq b_t^j, \quad 0 \leq q_t^r \leq b_t, \\ & q_t^j > 0 \Rightarrow p_t^j \leq \pi_t, \quad q_t^r > 0 \Rightarrow p_t \leq \pi_t. \end{aligned}$$

Thus, π_t is the **clearing price** satisfying market balance at minimum cost — i.e. the *merit-order price*. We can summarize it as a function:

$$\pi_t = f_t(p_t, b_t; \mathbf{p}_{-r,t}, \mathbf{b}_{-r,t}, D_t),$$

where $\mathbf{p}_{-r,t}, \mathbf{b}_{-r,t}$ are the other producers' bids.

⚡ 5. Market Acceptance and Allocation

Given the merit-order rule:

- Renewable producer's bid is **accepted** if $p_t \leq \pi_t$.
- Delivered quantity = b_t if accepted, else 0.

Define:

$$a_t(p_t, \pi_t) = 1_{\{p_t \leq \pi_t\}}.$$

💸 6. Delivery and Penalty Settlement

Actual generation g_t may differ from bid b_t, producing imbalances.

Define imbalance:

$$\delta_t = g_t - b_t.$$

Penalty cost (for both over- and under-delivery):

$$\text{Penalty}_t(g_t, b_t) = c_t^-(b_t - g_t)_+ + c_t^+(g_t - b_t)_+.$$

💰 7. Hourly Profit Function

Given clearing price π_t and generation g_t:

$$\Pi_t(p_t, b_t, g_t) = a_t(p_t, \pi_t) \left[p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+ \right].$$

🎯 8. Renewable Producer's Problem

The renewable producer chooses (b_t, p_t) to **maximize robust expected profit**, accounting for endogenous prices and deep uncertainty in generation.

Formally:

$$\max_{\{b_t,p_t\}_{t=1}^{24}} \quad \sum_{t=1}^{24} \min_{g_t \in \mathcal{G}_t} a_t(p_t, f_t(p_t, b_t; \mathbf{p}_{-r,t}, \mathbf{b}_{-r,t}, D_t)) \left[p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+\right]$$
$$\text{s.t.} \quad 0 \le b_t \le \bar{b}_t, \quad p_t \ge 0, \quad \forall t \in \mathcal{T}.$$

🧱 9. Model Components Summary

Type	Symbol	Description
Decision variables	b_t, p_t	Bid quantity and price (chosen by renewable producer)
Parameters	$D_t, (b_t^j, p_t^j), c_t^-, c_t^+, \bar{b}_t$	Demand, competitor bids, penalty rates, and capacity limit
State variables	none (single-day market)	
Uncertainties	$g_t \in \mathcal{G}_t$ (deep uncertainty)	
Endogenous variable	$\pi_t = f_t(p_t, b_t; \mathbf{p}_{-r,t}, \mathbf{b}_{-r,t}, D_t)$	
Indicator	$a_t = 1_{\{p_t \le \pi_t\}}$	
Payoff	$p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+$	
Transition function	none (24 independent hourly markets)	

🧮 10. Interpretation

- The renewable producer **anticipates** the merit-order price function $f_t(\cdot)$, given forecasts of demand and competitors' bids.
- The **profit** depends on whether the bid is accepted, the actual generation, and imbalance penalties.
- Because g_t's distribution is unknown, the producer **hedges against the worst-case realization** within the uncertainty set $\mathcal{G}_t$.
- This makes the optimization **distributionally robust** or **min–max** in structure.

🧠 11. Model Type

This is a **single-day, multi-hour, distributionally robust stochastic–equilibrium model**:

$$\text{Renewable producer: } \max_{b_t,p_t} \sum_t \min_{g_t \in \mathcal{G}_t} \Pi_t(p_t, b_t, g_t) \quad \text{subject to merit-order equilibrium constraints.}$$

If we embed the market-clearing condition explicitly for all producers, it becomes a **bilevel (or equilibrium) problem** — the upper level for the renewable producer's strategy, the lower level for the market operator's clearing.

Would you like me to extend this into an **explicit bilevel formulation** (upper level: renewable producer's bid optimization; lower level: system operator's market clearing)?
That would make the endogenous price mechanism mathematically precise and suitable for numerical modeling (e.g., MPEC formulation).

> You can assume the total energy demand D_t follows a normal distribution with constant mean and variance. You can also assume that for each conventional producer, its delivery quantity of energy and the offer price in bids follow normal distributions with constant means and variances. Please refine the model specification.

Perfect — now we can fully **close the market model**, since you've clarified that all other stochastic components (demand and conventional producers' bids) have **known probabilistic behavior**, while renewable generation remains **deeply uncertain**.

We can now combine all these elements into a **hybrid stochastic–robust bidding model** for the renewable producer.

✅ Refined Model Specification

---

1. Context and Objective

A renewable producer r participates in a **day-ahead merit-order market** for 24 hours of the next day.
For each hour $t = 1, \dots, 24$, the producer decides a bid:

- b_t: quantity (MWh),
- p_t: offer price ($/MWh).

The market clearing price π_t is determined by the **intersection of total supply (bids) and demand**, and thus depends on:

- stochastic **system demand** D_t,
- stochastic **competitor bids** (quantities and prices of conventional producers),
- the renewable producer's own bid (b_t, p_t).

After clearing:

- If the renewable's bid is accepted ($p_t \leq \pi_t$), it must deliver b_t;
- Its actual generation g_t (deeply uncertain) may differ, causing imbalance penalties for both under- and over-delivery.

The goal is to **maximize the expected profit** over all hours,
accounting for stochastic market prices and robustly handling renewable uncertainty.

2. Sets and Indices

Symbol	Description
$t \in \mathcal{T} = \{1, \dots, 24\}$	Hours of the next day
$j \in \mathcal{J} = \{1, 2, 3\}$	Conventional producers
r	Renewable producer (decision-maker)

3. Decision Variables (renewable producer's bids)

Symbol	Description
$b_t \geq 0$	Energy quantity offered (MWh)
$p_t \geq 0$	Offer price ($/MWh)

All decisions are made **day-ahead**.

4. Random Variables

4.1 Demand

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \quad \text{independent across } t.$$

4.2 Competitors' bids

For each producer $j \in \mathcal{J}$:

$$b_t^j \sim \mathcal{N}(\mu_{b,j}, \sigma_{b,j}^2), \quad p_t^j \sim \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2),$$

with independence across producers and hours.

These distributions are *known* (estimated from market data or historical behavior).

4.3 Renewable generation (deep uncertainty)

$$g_t \in \mathcal{G}_t = [\underline{g}_t, \overline{g}_t],$$

with **no known probability law**.
The renewable producer assumes the **worst-case realization** of g_t within $\mathcal{G}_t$.

5. Market Clearing Mechanism

The **system operator** clears the market hour-by-hour using a **merit-order rule**:

For each hour t:

- Sort all accepted offers (p_t, b_t) and (p_t^j, b_t^j) by ascending price;
- Determine the market clearing price π_t satisfying total supply = demand.

Formally, define total supply curve:

$$S_t(\pi) = b_t 1_{\{p_t \le \pi\}} + \sum_{j \in \mathcal{J}} b_t^j 1_{\{p_t^j \le \pi\}}.$$

Then the clearing price π_t satisfies:

$$S_t(\pi_t) = D_t.$$

Thus, π_t is a **random variable** determined by $(D_t, \{b_t^j, p_t^j\}_j)$ and the renewable producer's bid (b_t, p_t):

$$\pi_t = f_t(D_t, \{b_t^j, p_t^j\}_j; b_t, p_t).$$

---

6. Market Acceptance Indicator

$$a_t(p_t, \pi_t) = \begin{cases} 1, & \text{if } p_t \le \pi_t, \\ 0, & \text{otherwise.} \end{cases}$$

If $a_t = 0$: bid not accepted, profit = 0.
If $a_t = 1$: bid accepted, must deliver b_t.

---

7. Imbalance and Penalty Settlement

After real generation g_t is realized:

$$\delta_t = g_t - b_t.$$

Penalties apply symmetrically:

$$\text{Penalty}_t(g_t, b_t) = c_t^-(b_t - g_t)_+ + c_t^+(g_t - b_t)_+,$$

where
c_t^-: penalty for under-delivery,
c_t^+: penalty for over-delivery.

---

8. Hourly Profit Function

For given (p_t, b_t), realized π_t, and actual g_t:

$$\Pi_t(p_t, b_t; \pi_t, g_t) = a_t(p_t, \pi_t)\left[p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+\right].$$

---

9. Objective Function (Hybrid Stochastic–Robust Optimization)

The renewable producer seeks to **maximize expected profit over the stochastic market, robust to generation uncertainty**:

$$\max_{\{b_t, p_t\}_{t=1}^{24}} \quad \mathbb{E}_{D_t, \{b_t^j, p_t^j\}} \left[\sum_{t=1}^{24} \min_{g_t \in \mathcal{G}_t} a_t(p_t, f_t(D_t, \{b_t^j, p_t^j\}_j; b_t, p_t)) \big(p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+ \big) \right]$$
$$\text{s.t.} \quad 0 \le b_t \le \bar{b}_t, \quad p_t \ge 0, \quad \forall t \in \mathcal{T}.$$

10. Model Components Summary

Type	Symbol	Description
Decision variables	b_t, p_t	Bid quantity and offer price (renewable producer)
Random variables (known distributions)	D_t, b_t^j, p_t^j	Demand and competitors' bids (Normal)
Deep uncertainty	$g_t \in [\underline{g}_t, \overline{g}_t]$	Renewable generation (unknown distribution)
Parameters	$c_t^-, c_t^+, \bar{b}_t$	Penalty rates and capacity
Market function	$f_t(D_t, \{b_t^j, p_t^j\}_j; b_t, p_t)$	Merit-order clearing price
Acceptance rule	$a_t = 1_{\{p_t \le \pi_t\}}$	Determines bid acceptance
Profit function	$p_t b_t - c_t^-(b_t - g_t)_+ - c_t^+(g_t - b_t)_+$	If bid accepted
Objective	$\mathbb{E}[\min_{g_t \in \mathcal{G}_t} \sum_t \Pi_t]$	Expected profit under market uncertainty, robust to generation uncertainty

11. Interpretation

- **Stochastic component**:
  The renewable producer treats the market clearing price π_t as a random variable induced by stochastic demand and competitors' bids.
  The expectation operator $\mathbb{E}[\cdot]$ averages over this known uncertainty.
- **Robust component**:
  Because renewable output g_t lacks a known distribution, the producer uses a **worst-case (min)** operator over the feasible range $\mathcal{G}_t$.
- **Market coupling**:
  The endogenous function $f_t(\cdot)$ links the renewable bid to the market price, reflecting strategic interdependence with conventional competitors.

12. Optional Simplifications for Computation

Depending on the intended analysis:

1. **Price Distribution Approximation**
   If the merit-order clearing is too complex to model directly, approximate π_t as a stochastic variable

with mean and variance derived from D_t and competitors' distributions.

2. **Scenario-based Approximation**
   Sample scenarios of (D_t, b_t^j, p_t^j) and solve a scenario-based **stochastic–robust optimization** problem.
3. **Risk-averse Variant**
   Replace the expectation with a **CVaR** or **risk-weighted** measure to incorporate the producer's risk attitude.

✅ Final Form (Compact)

$$\boxed{\max_{b_t, p_t} \mathbb{E}_{D_t, \{b_t^j, p_t^j\}} \left[\sum_{t=1}^{24} \min_{g_t \in [\underline{g}_t, \overline{g}_t]} 1_{\{p_t \le f_t(D_t, \{b_t^j, p_t^j\}; b_t, p_t)\}} \left(p_t b_t - c_t^- (b_t - g_t)_+ - c_t^+ (g_t - b_t)_+ \right) \right]}$$

subject to $0 \le b_t \le \bar{b}_t, \; p_t \ge 0.$

Would you like me to take the next step and **derive an explicit functional form** or **approximation for the clearing price** $f_t(\cdot)$ under normal assumptions (e.g., when total supply and demand curves are linear)?
That would make the model fully solvable or simulatable.

> Various stakeholders approach this problem from different perspectives. Please identify four different perspectives related to this problem. These perspectives should share a common environment, along with global parameters, state variables, and an initial state. For each perspective, also provide the corresponding model with the key components, such as control variables, transition functions, and objective functions. If there are multiple objectives, provide multiple objective functions instead of a weighted objective function.

Excellent — this is now moving from *one agent's optimization problem* (the renewable producer) to a **multi-perspective system-level modeling framework**.

We'll structure the environment as a **shared market–system environment** with common parameters and state variables, then formalize **four distinct stakeholder perspectives**, each with its own objectives, controls, and transitions.

🌍 Common Environment

All four perspectives share the same **market environment**, **time horizon**, **parameters**, and **state representation**.

---

1. Shared Environment

Time Horizon

- $t \in \mathcal{T} = \{1, \dots, 24\}$: hourly periods in the next day.

State Variables

Symbol	Description
D_t	System demand (stochastic, $\mathcal{N}(\mu_D, \sigma_D^2)$)
g_t	Renewable generation (deep uncertainty, $g_t \in [\underline{g}_t, \overline{g}_t]$)
π_t	Market clearing price (endogenous)
$\mathbf{b}_t = \{b_t^r, b_t^1, b_t^2, b_t^3\}$	Bid quantities from all producers
$\mathbf{p}_t = \{p_t^r, p_t^1, p_t^2, p_t^3\}$	Offer prices from all producers
S_t	System supply function $S_t(\pi) = \sum_i b_t^i 1_{\{p_t^i \leq \pi\}}$

Global Parameters

Symbol	Description
μ_D, σ_D^2	Demand mean and variance
$(\mu_{b,j}, \sigma_{b,j}^2), (\mu_{p,j}, \sigma_{p,j}^2)$	Conventional producers' bid parameters
c_t^-, c_t^+	Penalty rates for renewable imbalance
$\bar{b}_t^r$	Renewable capacity
$N_p = 4$	Total number of producers
Ω	Scenario space for stochastic variables
$f_t(\cdot)$	Market clearing function (merit order rule)

Initial State (before bidding)

$$s_0 = \left\{ D_t \sim \mathcal{N}(\mu_D, \sigma_D^2),\ b_t^j \sim \mathcal{N}(\mu_{b,j}, \sigma_{b,j}^2),\ p_t^j \sim \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2),\ g_t \in [\underline{g}_t, \overline{g}_t] \right\}.$$

👥 Four Stakeholder Perspectives

We now define **four perspectives**, each with its own decision variables (controls), transitions, and objectives — all operating within the above environment.

Perspective 1: Renewable Energy Producer

Role

Strategic bidder facing uncertain generation and prices.

Control Variables

- b_t^r: bid quantity (MWh)
- p_t^r: offer price ($/MWh)

Transition Function

- Market price update:

$$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t)$$

- Generation realization:

$$g_t \in [\underline{g}_t, \overline{g}_t]$$

(no temporal coupling → static per hour)

Objective Function(s)

Two objectives typically matter:

1. **Expected Profit Maximization**

$$\max \ \mathbb{E}_{D_t, \{b_t^j, p_t^j\}} \left[\sum_{t=1}^{24} a_t(p_t^r, \pi_t)(p_t^r b_t^r - c_t^- (b_t^r - g_t)_+ - c_t^+ (g_t - b_t^r)_+) \right]$$

   with $a_t = 1_{\{p_t^r \le \pi_t\}}$.

2. **Reliability Objective (Minimize imbalance exposure)**

$$\min \sum_{t=1}^{24} \max_{g_t \in [\underline{g}_t, \overline{g}_t]} |b_t^r - g_t|.$$

Interpretation

A **stochastic–robust optimization** problem balancing profit vs. reliability risk under market and generation uncertainty.

Perspective 2: Conventional Producer

Role

Price-taking or strategic participant in the same day-ahead market with predictable output and costs.

Control Variables

- b_t^j: bid quantity (MWh)
- p_t^j: offer price ($/MWh)

Transition Function

- Market price formation:

$$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t)$$

- Dispatch quantity:

$$q_t^j = b_t^j \mathbf{1}_{\{p_t^j \le \pi_t\}}$$

Objective Function(s)

1. **Expected Profit Maximization**

$$\max \ \mathbb{E}_{D_t, \mathbf{b}_{-j,t}, \mathbf{p}_{-j,t}} \left[\sum_{t=1}^{24} (p_t^j - C_j'(q_t^j)) q_t^j \right],$$

   where $C_j'(q_t^j)$ is the marginal generation cost.

2. **Market Share Stabilization**

$$\min \sum_{t=1}^{24} (q_t^j - \bar{q}_j)^2,$$

   where $\bar{q}_j$ is a target output level (to avoid volatility).

Interpretation

A **stochastic optimization problem** over market uncertainties; decisions impact the merit-order and thus prices.

Perspective 3: System Operator (Market Operator)

Role

Clears the market to ensure demand-supply balance at minimum system cost, given submitted bids and demand uncertainty.

Control Variables

- π_t: clearing price ($/MWh)
- q_t^i: accepted quantity for each producer $i \in \{r, 1, 2, 3\}$

Transition Function

- Market equilibrium (merit order):

$$\sum_i q_t^i = D_t, \quad 0 \le q_t^i \le b_t^i, \quad q_t^i > 0 \Rightarrow p_t^i \le \pi_t.$$

- Prices update as $\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t)$.

Objective Function(s)

1. **System Cost Minimization**

$$\min \sum_{t=1}^{24} \sum_{i} p_t^i q_t^i \quad \text{s.t. balance constraints.}$$

2. **Market Efficiency (Minimize price volatility)**

$$\min \sum_{t=2}^{24} (\pi_t - \pi_{t-1})^2.$$

Interpretation

A **central clearing problem** ensuring efficient market operation and smooth price dynamics under uncertainty.

Perspective 4: Regulator / Social Planner

Role

Oversees the market to promote economic efficiency, environmental sustainability, and reliability.

Control Variables

- Policy levers such as:
  - τ_t: renewable subsidy or carbon tax ($/MWh)
  - $c_t^{\pm}$: imbalance penalty settings
  - R_t: renewable reserve requirement

Transition Function

- Policy affects producer behavior and hence equilibrium prices:

$$\pi_t' = f_t(D_t, \mathbf{b}_t(\tau_t), \mathbf{p}_t(\tau_t)).$$

Objective Function(s)

1. **Social Welfare Maximization**

$$\max \mathbb{E}\left[\sum_{t=1}^{24} (U(D_t) - \sum_{i} p_t^i q_t^i)\right],$$

where $U(D_t)$ is consumer utility.

2. **Emission Reduction Objective**

$$\min \sum_{t=1}^{24} \text{Emissions}_t = \sum_{t} \sum_{j \in \mathcal{J}} \alpha_j q_t^j,$$

with emission factors α_j.

OpenAI

3. **Reliability Objective**

$$\min \sum_{t=1}^{24} (D_t - \sum_i q_t^i)_+.$$

Interpretation

A **multi-objective regulatory optimization** problem: maximize welfare, minimize emissions, and maintain reliability by adjusting market rules and penalties.

🔗 Summary Table of the Four Perspectives

Perspective	Decision Variables	Transition Function	Objectives
1. Renewable Producer	b_t^r, p_t^r	$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t); g_t \in [\underline{g}_t, \overline{g}_t]$	(i) Max expected profit; (ii) Min imbalance exposure
2. Conventional Producer	b_t^j, p_t^j	$\pi_t = f_t(\cdot); q_t^j = b_t^j 1_{\{p_t^j \le \pi_t\}}$	(i) Max expected profit; (ii) Min output variance
3. System Operator	π_t, q_t^i	$\sum_i q_t^i = D_t; q_t^i > 0 \Rightarrow p_t^i \le \pi_t$	(i) Min total cost; (ii) Min price volatility
4. Regulator / Social Planner	$\tau_t, c_t^{\pm}, R_t$	$\pi_t' = f_t(D_t, \mathbf{b}_t(\tau_t), \mathbf{p}_t(\tau_t))$	(i) Max social welfare; (ii) Min emissions; (iii) Min unmet demand

---

🧠 Interpretation

- All four actors share the **same system environment** and **state evolution**, but differ in **control authority** and **objectives**.
- Their models collectively form a **hierarchical stochastic–robust market ecosystem**:
  - Producers (renewable and conventional) → **micro-level** optimizers.
  - System Operator → **market equilibrium** enforcer.
  - Regulator → **macro-level** policy optimizer.
- The full system could be represented as a **multi-level stochastic game** or **hierarchical optimization structure**, with each layer responding to or influencing the others.

Would you like me to write out the **mathematical bilevel hierarchy** (i.e., explicit Stackelberg game or equilibrium system combining these four perspectives)? That would show exactly how the regulator, operator, and producers interact within one unified mathematical model.

The second and third perspectives are inconsistent with the problem description. The bid quantities and bid prices of conventional producers follow normal distributions with constant means and variances. Similarly, the clearing price and the accepted quantity for each producer are determined based on the merit order. These variables cannot be used as control variables. Please refine the model specification.

Excellent — that's exactly right.

Since the **conventional producers' bids** and **demand** are stochastic but **not controlled** (they follow known distributions), and since the **system operator** only clears the market *mechanically* using the merit-order rule (not through optimization decisions), the only true decision-maker here is the **renewable producer**, while the other perspectives describe **different evaluative or monitoring roles** in the same environment.

We can therefore revise the specification to create **four logically consistent stakeholder perspectives**, all sharing the same environment, parameters, and states — but differing in what they *control*, *observe*, or *optimize*.

🌍 Common Environment (Shared by All Perspectives)

All stakeholders share the same 24-hour day-ahead electricity market environment.

Time Horizon

$$t \in \mathcal{T} = \{1, \dots, 24\}.$$

Global Parameters

Symbol	Description
μ_D, σ_D^2	Mean and variance of demand
$\mu_{b,j}, \sigma_{b,j}^2$	Mean and variance of bid quantities for each conventional producer j
$\mu_{p,j}, \sigma_{p,j}^2$	Mean and variance of bid prices for each conventional producer j
c_t^-, c_t^+	Penalty rates for under- and over-delivery
$\bar{b}_t^r$	Maximum renewable bid (capacity limit)
$N_p = 4$	Number of producers (1 renewable + 3 conventional)

Stochastic Variables

Symbol	Distribution	Description
$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$	Total system demand	
$b_t^j \sim \mathcal{N}(\mu_{b,j}, \sigma_{b,j}^2)$	Conventional producer j's bid quantity	
$p_t^j \sim \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2)$	Conventional producer j's offer price	
$g_t \in [\underline{g}_t, \overline{g}_t]$	Renewable generation (deeply uncertain)	

State Variables

Symbol	Description
D_t	Demand realization

Symbol	Description
$\mathbf{b}_t = \{b_t^r, b_t^1, b_t^2, b_t^3\}$	Bid quantities
$\mathbf{p}_t = \{p_t^r, p_t^1, p_t^2, p_t^3\}$	Offer prices
π_t	Market clearing price (determined by merit order)
q_t^i	Accepted delivery for producer $i \in \{r, 1, 2, 3\}$

Market Clearing (Merit Order Rule)

For each hour t:

$$S_t(\pi) = \sum_{i \in \{r,1,2,3\}} b_t^i \, 1_{\{p_t^i \le \pi\}},$$

and the market clearing price satisfies:

$$S_t(\pi_t) = D_t.$$

Accepted quantities:

$$q_t^i = b_t^i \, 1_{\{p_t^i \le \pi_t\}}.$$

Initial State

$$s_0 = \{D_t, g_t, \mathbf{b}_t, \mathbf{p}_t \ (\text{uncertain for all } t)\}.$$

👥 Four Consistent Stakeholder Perspectives

---

1. Renewable Producer (Decision-Maker)

Role:
Strategically bids quantities and prices to maximize profit under market and renewable uncertainty.

Control Variables

- $b_t^r \ge 0$: offered energy quantity (MWh)
- $p_t^r \ge 0$: offer price ($/MWh)

Transition Function

- Market clearing:

$$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t), \quad q_t^r = b_t^r \, 1_{\{p_t^r \le \pi_t\}}.$$

- Generation realization:

$$g_t \in [\underline{g}_t, \overline{g}_t].$$

Objective Functions

1. **Expected Profit Maximization:**

$$\max_{b_t^r, p_t^r} \mathbb{E}_{D_t, \{b_t^j, p_t^j\}} \left[\sum_{t=1}^{24} 1_{\{p_t^r \le \pi_t\}} (p_t^r b_t^r - c_t^- (b_t^r - g_t)_+ - c_t^+ (g_t - b_t^r)_+) \right].$$

2. **Robustness to Generation Uncertainty:**

$$\min_{g_t \in [\underline{g}_t, \overline{g}_t]} \sum_{t=1}^{24} |b_t^r - g_t|.$$

---

2. Conventional Producers (Stochastic Actors)

Role:
Their bids are random variables that influence price formation but are not controlled by them in this model (treated as exogenous stochastic processes).

Control Variables

- None (since b_t^j, p_t^j follow known normal distributions).

Transition Function

- Same market clearing function:

$$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t), \quad q_t^j = b_t^j \, 1_{\{p_t^j \le \pi_t\}}.$$

Objective Functions (Analytical / Evaluative, not optimization):

1. **Expected Profit (given distributions):**

$$\mathbb{E}[\Pi_t^j] = \mathbb{E}_{D_t, \mathbf{b}_t, \mathbf{p}_t} \left[(p_t^j - C_j'(b_t^j)) b_t^j \, 1_{\{p_t^j \le \pi_t\}} \right].$$

2. **Reliability (Probability of Acceptance):**

$$P(a_t^j = 1) = P(p_t^j \le \pi_t).$$

These are *performance metrics* rather than control objectives — reflecting their probabilistic influence on the market outcome.

---

3. System Operator (Market Clearing Mechanism)

Role:
Clears the market each hour based on merit order; has no control variables beyond implementing the clearing rule.

Control Variables

- None (the operator executes the deterministic rule).

Transition Function (Defining the Price)

$$\pi_t = \inf\{\pi : S_t(\pi) \geq D_t\}.$$

$$q_t^i = b_t^i \, 1_{\{p_t^i \leq \pi_t\}}.$$

Objective Functions (Analytical Performance Metrics)

1. **Market Balance:**

$$\sum_i q_t^i = D_t.$$

   (Constraint, always enforced by merit order.)

2. **Market Efficiency Metric:**

$$\text{Total Cost}_t = \sum_i p_t^i q_t^i,$$

   which can be tracked to evaluate economic efficiency but not optimized.

---

4. Regulator / Policy Maker (Oversight Role)

Role:
Designs policy levers to ensure efficient, reliable, and sustainable market outcomes.
Although they don't control hourly market variables, they can adjust *global parameters* like penalties or subsidies.

Control Variables

- c_t^-, c_t^+: imbalance penalty rates
- τ_t: renewable subsidy or tax
- R_t: renewable share target or reserve margin

Transition Function

- Adjusted payoff and bid behavior:

$$\text{Renewable Profit}_t' = (p_t^r + \tau_t) b_t^r - c_t^- (b_t^r - g_t)_+ - c_t^+ (g_t - b_t^r)_+.$$

- Market equilibrium remains merit-order based:

$$\pi_t' = f_t(D_t, \mathbf{b}_t(\tau_t), \mathbf{p}_t(\tau_t)).$$

Objective Functions (Multiple)

1. **Social Welfare:**

$$\max \mathbb{E}\left[\sum_{t=1}^{24}(U(D_t) - \sum_i p_t^i q_t^i)\right].$$

2. **Emission Reduction:**

$$\min \sum_{t=1}^{24} \sum_{j\in\{1,2,3\}} \alpha_j q_t^j.$$

3. **Reliability:**

$$\min \sum_{t=1}^{24}(D_t - \sum_i q_t^i)_+.$$

🔗 Summary of Perspectives

Perspective	Role	Control Variables	Transition Function	Objectives
1. Renewable Producer	Strategic bidder	b_t^r, p_t^r	Merit order $\to \pi_t, q_t^r$; generation realization g_t	(1) Max expected profit (2) Min imbalance
2. Conventional Producers	Stochastic market participants	None (bids ~ Normal)	Merit order $\to \pi_t, q_t^j$	(1) Expected profit (2) Acceptance probability
3. System Operator	Market-clearing mechanism	None (mechanical)	$S_t(\pi_t) = D_t, q_t^i = b_t^i 1_{\{p_t^i \le \pi_t\}}$	(1) Market balance (2) Efficiency metric
4. Regulator / Policy Maker	Oversees market design	$c_t^-, c_t^+, \tau_t, R_t$	Policies influence bids → new equilibrium	(1) Max social welfare (2) Min emissions (3) Min unmet demand

---

🧠 Conceptual Interpretation

- **Common environment:** stochastic–robust day-ahead electricity market.
- **Actors share the same state variables** (D_t, g_t, π_t, q_t^i), but differ in what they influence or observe.
- **Only the renewable producer** directly chooses bid variables.
- The **system operator** and **conventional producers** have *no decision control* in this setup — their behaviors are modeled as either deterministic mechanisms or stochastic processes.
- The **regulator** acts on a *slower timescale*, adjusting global parameters that shape all market outcomes.

Would you like me to now show how these four perspectives can be linked together in a **multi-layer market system diagram** (with causal and informational flows between stakeholders)? That would make the structure of interactions visually and mathematically explicit.

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 controls and objectives. The composed model should: 1. embed each perspective's control variables and objectives; 2. enables modular evaluation of each perspective; 3. for each perspective, treat other perspectives' controls as exogenous inputs; and 4. incorporate all other components from every perspective.

Excellent — this is the final and most complete step: constructing a **unified, internally consistent model** that (1) shares a common environment and state dynamics, (2) embeds each stakeholder's control variables and objectives, and (3) allows modular evaluation where each actor treats the others' controls as given.

What we're building is a **multi-perspective, modular stochastic–robust market system model**: a single mathematical framework that encompasses the renewable producer, conventional producers, system operator, and regulator, each with their own controls, objectives, and dependencies — all interacting through a shared state evolution.

🧩 1. Unified Model Overview

We define the **shared environment** and the **perspective-specific submodels** as components of one integrated system:

$$\mathcal{M} = (\mathcal{E}, \mathcal{S}, \mathcal{C}, \mathcal{O}),$$

where:

- $\mathcal{E}$: shared environment and parameters,
- $\mathcal{S}$: common state variables and transition functions,
- $\mathcal{C}$: control variables by perspective,
- $\mathcal{O}$: perspective-specific objective functions.

Each perspective (renewable, conventional, operator, regulator) is a **module** $m_k = (\mathcal{C}_k, \mathcal{O}_k)$ acting on the same environment and states.

🌍 2. Shared Environment $\mathcal{E}$

Time Horizon

$$t \in \mathcal{T} = \{1, \dots, 24\}.$$

Global Parameters

Symbol	Description
μ_D, σ_D^2	Mean and variance of system demand
$(\mu_{b,j}, \sigma_{b,j}^2), (\mu_{p,j}, \sigma_{p,j}^2)$	Mean and variance of conventional producers' bids
c_t^-, c_t^+	Penalty rates for imbalance
$\bar{b}_t^r$	Max renewable bid (capacity)
α_j	Emission factor for conventional producer j
$U(D_t)$	Consumer utility from energy consumption
τ_t	Renewable subsidy (if active)
R_t	Renewable target / reserve margin

Random and Deeply Uncertain Variables

Symbol	Distribution / Set	Description
$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$	Stochastic demand	
$b_t^j \sim \mathcal{N}(\mu_{b,j}, \sigma_{b,j}^2)$	Conventional bid quantity	
$p_t^j \sim \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2)$	Conventional offer price	
$g_t \in [\underline{g}_t, \overline{g}_t]$	Renewable generation (deep uncertainty)	

🧮 3. Common State Space $\mathcal{S}$

At each hour t, define the system state:

$$s_t = (D_t, g_t, \mathbf{b}_t, \mathbf{p}_t, \pi_t, \mathbf{q}_t),$$

where:

- D_t: realized demand,
- g_t: renewable generation,
- $\mathbf{b}_t = \{b_t^r, b_t^1, b_t^2, b_t^3\}$: bid quantities,
- $\mathbf{p}_t = \{p_t^r, p_t^1, p_t^2, p_t^3\}$: offer prices,
- π_t: market clearing price,
- $\mathbf{q}_t = \{q_t^r, q_t^1, q_t^2, q_t^3\}$: accepted quantities.

Transition Functions (Shared Mechanisms)

(a) Market Clearing (Merit Order):

$$S_t(\pi) = \sum_{i \in \{r,1,2,3\}} b_t^i \, 1_{\{p_t^i \le \pi\}}, \quad \text{and} \quad S_t(\pi_t) = D_t.$$

Clearing price and dispatch:

$$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t), \quad q_t^i = b_t^i \, 1_{\{p_t^i \le \pi_t\}}.$$

(b) Generation Realization:

$$g_t \in [\underline{g}_t, \overline{g}_t].$$

(c) State Transition (hourly independence):

$$s_{t+1} = f(s_t, \mathbf{C}_t, \text{random draws for } D_{t+1}, b_{t+1}^j, p_{t+1}^j, g_{t+1}).$$

Initial state s_0: prior distributions of D_t, b_t^j, p_t^j, g_t.

🎛️ 4. Control Variables $\mathcal{C}$ by Perspective

Perspective	Control Variables	Treated as Exogenous by Others
Renewable Producer	b_t^r, p_t^r	Yes, for all others
Conventional Producers	None (stochastic variables)	Yes
System Operator	None (mechanical clearing)	Yes
Regulator / Policy Maker	$c_t^-, c_t^+, \tau_t, R_t$	Yes, fixed for producers

All actors observe the same environment and state, but control disjoint subsets of parameters or actions.

🎯 5. Objective Functions $\mathcal{O}$

Each perspective has distinct goals, all evaluated within the same environment and state transitions.

(a) Renewable Producer $m_r = (\mathcal{C}_r, \mathcal{O}_r)$

Controls

$$\mathcal{C}_r = \{b_t^r, p_t^r\}.$$

Profit Function

$$\Pi_t^r = 1_{\{p_t^r \le \pi_t\}} \left[(p_t^r + \tau_t) b_t^r - c_t^- (b_t^r - g_t)_+ - c_t^+ (g_t - b_t^r)_+ \right].$$

Objectives

1. **Expected Profit Maximization:**

$$\max_{b_t^r, p_t^r} \mathbb{E}_{D_t, \{b_t^j, p_t^j\}} \left[\sum_{t=1}^{24} \Pi_t^r \right].$$

2. **Reliability (Minimize Imbalance Magnitude):**

$$\min_{b_t^r, p_t^r} \sum_{t=1}^{24} \max_{g_t \in [\underline{g}_t, \overline{g}_t]} |b_t^r - g_t|.$$

(b) Conventional Producers $m_c = (\mathcal{C}_c, \mathcal{O}_c)$

Controls

None — bids are stochastic:

$$b_t^j \sim \mathcal{N}(\mu_{b,j}, \sigma_{b,j}^2), \quad p_t^j \sim \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2).$$

Derived Quantities

$$q_t^j = b_t^j \, 1_{\{p_t^j \le \pi_t\}}.$$

Objectives (Evaluative)

1. **Expected Profit (given distributions):**

$$\mathbb{E}[\Pi_t^j] = \mathbb{E}[(p_t^j - C_j'(b_t^j)) b_t^j \, 1_{\{p_t^j \le \pi_t\}}].$$

2. **Acceptance Probability:**

$$P(a_t^j = 1) = P(p_t^j \le \pi_t).$$

These are *evaluations* rather than optimization objectives.

(c) System Operator $m_o = (\mathcal{C}_o, \mathcal{O}_o)$

Controls

None — market-clearing is a rule, not a decision.

Transition / Mechanism

Implements:

$$\pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t), \quad q_t^i = b_t^i \, 1_{\{p_t^i \le \pi_t\}}.$$

Objectives (Analytical Evaluation)

1. **Market Balance:**

$$\sum_i q_t^i = D_t.$$

2. **System Efficiency Metric:**

$$\text{Total Cost}_t = \sum_i p_t^i q_t^i.$$

3. **Price Stability (if monitored):**

$$\min \sum_{t=2}^{24} (\pi_t - \pi_{t-1})^2.$$

(Used for evaluation, not active optimization.)

(d) Regulator / Policy Maker $m_R = (\mathcal{C}_R, \mathcal{O}_R)$

Controls

$$\mathcal{C}_R = \{c_t^-, c_t^+, \tau_t, R_t\}.$$

Influence

Policy parameters modify renewable payoffs and equilibrium outcomes:

$$\pi_t' = f_t(D_t, \mathbf{b}_t(\tau_t), \mathbf{p}_t(\tau_t)).$$

Objectives

1. **Social Welfare Maximization:**

$$\max_{\mathcal{C}_R} \mathbb{E}\left[\sum_{t=1}^{24} (U(D_t) - \sum_i p_t^i q_t^i)\right].$$

2. **Emission Minimization:**

$$\min_{\mathcal{C}_R} \sum_{t=1}^{24} \sum_{j \in \{1,2,3\}} \alpha_j q_t^j.$$

3. **Reliability (Energy Adequacy):**

$$\min_{\mathcal{C}_R} \sum_{t=1}^{24} (D_t - \sum_i q_t^i)_+.$$

🧠 6. Composed Model Structure (Modular Integration)

We now assemble all modules into the **Unified Market System**:

Unified Model $\mathcal{M}^* = \{\mathcal{E}, \mathcal{S}, m_r, m_c, m_o, m_R\}$

Each module is evaluated *within the same environment and states* but has distinct control and objective sets.

(a) Shared State Dynamics

$$\begin{cases} D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \\ b_t^j, p_t^j \sim \mathcal{N}(\mu_{b,j}, \sigma_{b,j}^2), \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2), \\ g_t \in [\underline{g}_t, \overline{g}_t], \\ \pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t), \\ q_t^i = b_t^i \, 1_{\{p_t^i \leq \pi_t\}}. \end{cases}$$

(b) Module Evaluations

$$\begin{aligned} \text{Renewable module:} \quad & (\mathcal{C}_r, \mathcal{O}_r) \text{ as above.} \\ \text{Conventional modules:} \quad & (\mathcal{C}_c, \mathcal{O}_c) \text{ as above.} \\ \text{System operator:} \quad & (\mathcal{C}_o, \mathcal{O}_o). \\ \text{Regulator:} \quad & (\mathcal{C}_R, \mathcal{O}_R). \end{aligned}$$

(c) Cross-Module Interactions

From	To	Effect
Regulator	Renewable Producer	Alters penalty rates $c_t^{\pm}$ and subsidy τ_t
Renewable Producer	System Operator	Affects price formation via b_t^r, p_t^r
Conventional Producers	System Operator	Affect π_t stochastically
System Operator	All	Determines π_t, q_t^i, influencing all profits and metrics
Regulator	All	Evaluates global welfare, emissions, reliability

Each module takes other actors' controls as **exogenous inputs**.

🔄 7. Properties of the Unified Model

Property	Description
Shared environment	Common parameters, uncertainties, and market-clearing rules.
Modular evaluation	Each perspective can be simulated or optimized independently while using the same state transitions.
Exogenous treatment	Each actor treats others' decisions or parameters as given.
Composability	The model can be solved sequentially (nested optimization) or evaluated jointly (equilibrium analysis).
Multi-objective ready	Each module supports multiple, separate objectives (no aggregation needed).

✅ 8. Compact System Representation

$$\begin{array}{rl} \textbf{Environment:} & \mathcal{E}, \quad s_t = (D_t, g_t, \mathbf{b}_t, \mathbf{p}_t, \pi_t, \mathbf{q}_t). \\ \textbf{Transitions:} & \pi_t = f_t(D_t, \mathbf{b}_t, \mathbf{p}_t), \quad q_t^i = b_t^i 1_{\{p_t^i \le \pi_t\}}. \\ \textbf{Modules:} & \begin{cases} m_r : \text{Renewable producer, controls } (b_t^r, p_t^r), \text{ objectives } \mathcal{O}_r, \\ m_c : \text{Conventional producers, stochastic, objectives } \mathcal{O}_c, \\ m_o : \text{System operator, no controls, objectives } \mathcal{O}_o, \\ m_R : \text{Regulator, controls } (c_t^{\pm}, \tau_t, R_t), \text{ objectives } \mathcal{O}_R. \end{cases} \end{array}$$

🧭 9. Interpretation

- This unified model describes **one coherent energy market ecosystem** with multiple perspectives operating over the same states and uncertainties.
- Each module can be **evaluated modularly** (e.g., simulate renewable profits given policy parameters, or assess welfare under new penalties).
- Cross-perspective interactions are clearly defined through shared variables (price, accepted quantities, demand).
- This formulation supports either **nested analysis** (e.g., regulator sets parameters → renewable optimizes bids → market clears) or **equilibrium evaluation** (all actors' strategies consistent).

Would you like me to express this unified system next as a **hierarchical game-theoretic formulation** (e.g., a Stackelberg or multi-level stochastic game), where the regulator is the upper-level leader, the renewable producer is a follower, and the operator enforces equilibrium constraints? That would make the architecture directly ready for analytical or computational treatment.

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 accept a dictionary of control variables and a dictionary of uncertain parameter values as input, simulate the model dynamics, and return the value of each objective function as output. This function does not need to be executed.

Here's a **modular Python implementation** of the unified market model that you requested.
Each stakeholder perspective is encapsulated as a **class** with its own controls, objectives, and evaluative methods — all sharing the same environment and market state logic.

At the end, a single **interface function** `simulate_market()` accepts:

- a dictionary of **control variables** (e.g., renewable bids, regulatory parameters), and
- a dictionary of **uncertain parameters** (realizations or sample draws of demand, generation, etc.), then simulates the shared market-clearing mechanism and evaluates each actor's objectives.

```python
import numpy as np

# ================================================================
# 1. Shared Environment and State Definitions
# ================================================================

class MarketEnvironment:
    """Shared market environment and stochastic inputs."""
    def __init__(self, params, uncertainties):
        """
        params: dict of global constants
        uncertainties: dict of uncertain variables or samples
        """
        self.params = params
        self.uncertainties = uncertainties
        self.num_hours = 24

    def get_demand(self, t):
        mu, sigma = self.params["mu_D"], self.params["sigma_D"]
        if "D_t" in self.uncertainties:
            return self.uncertainties["D_t"][t]
        return np.random.normal(mu, sigma)

    def get_conventional_bids(self, t):
        """Return random bids for the 3 conventional producers."""
        bids = []
        for j in range(3):
            mu_b, sig_b = self.params["mu_b"][j], self.params["sigma_b"][j]
            mu_p, sig_p = self.params["mu_p"][j], self.params["sigma_p"][j]
            b = (self.uncertainties.get("b_t^j", [None]*3)[j]
                 or np.random.normal(mu_b, sig_b))
            p = (self.uncertainties.get("p_t^j", [None]*3)[j]
                 or np.random.normal(mu_p, sig_p))
            bids.append((b, p))
        return bids

    def get_generation(self, t):
        g_min, g_max = self.params["g_bounds"]
        if "g_t" in self.uncertainties:
            return self.uncertainties["g_t"][t]
        return np.random.uniform(g_min, g_max)


# ================================================================
# 2. Market Operator: Merit-Order Clearing
# ================================================================

class SystemOperator:
    """Implements the market-clearing (merit-order) mechanism."""
    def __init__(self, env):
        self.env = env

    def clear_market(self, bids, demand):
        """
        bids: list of (b_i, p_i) tuples for all producers [renewable + 3 conventional]
        demand: scalar demand D_t
        Returns: clearing price π_t and accepted quantities q_i for each producer.
        """
        # Sort bids by price (ascending merit order)
        bids_sorted = sorted(enumerate(bids), key=lambda x: x[1][1])
        supply = 0.0
        clearing_price = None
        accepted = [0.0] * len(bids)

        for idx, (b, p) in [(i, bid) for i, bid in bids_sorted]:
            if supply + b < demand:
                supply += b
                accepted[idx] = b
            else:
                # Marginal bidder
                accepted[idx] = max(0, demand - supply)
                clearing_price = p
```

```
                break

        # If all bids used and still below demand, set highest price as clearing
        if clearing_price is None:
            clearing_price = max(p for (_, p) in bids)

        return clearing_price, accepted


# ================================================================
# 3. Renewable Producer
# ================================================================

class RenewableProducer:
    def __init__(self, env):
        self.env = env
        self.controls = {"b_r": np.zeros(24), "p_r": np.zeros(24)}

    def set_controls(self, controls):
        self.controls["b_r"] = np.array(controls.get("b_r", np.zeros(24)))
        self.controls["p_r"] = np.array(controls.get("p_r", np.zeros(24)))

    def profit_objective(self, prices, accepted, generation):
        """Compute expected profit over 24 hours given price outcomes."""
        c_minus, c_plus = self.env.params["c_minus"], self.env.params["c_plus"]
        tau = self.env.params.get("tau", np.zeros(24))
        profit = 0.0
        for t in range(24):
            a_t = 1.0 if accepted[t] > 0 else 0.0
            b_t, p_t = self.controls["b_r"][t], self.controls["p_r"][t]
            g_t = generation[t]
            profit_t = a_t * ((p_t + tau[t]) * b_t
                              - c_minus * max(b_t - g_t, 0)
                              - c_plus * max(g_t - b_t, 0))
            profit += profit_t
        return profit

    def imbalance_objective(self, generation):
        """Compute imbalance exposure."""
        imbalance = np.sum(np.abs(self.controls["b_r"] - generation))
        return imbalance


# ================================================================
# 4. Conventional Producers (stochastic participants)
# ================================================================

class ConventionalProducers:
    """Represent the three conventional producers."""
    def __init__(self, env):
        self.env = env

    def expected_profit_metrics(self, prices, accepted_bids):
        """Return expected profit and acceptance probabilities (evaluation only)."""
        results = []
        for j in range(3):
            mu_p = self.env.params["mu_p"][j]
            mu_b = self.env.params["mu_b"][j]
            Cj = self.env.params.get("cost_j", [0, 0, 0])[j]
            # Approximate average accepted value
            profit = (mu_p - Cj) * mu_b * np.mean([1 if a > 0 else 0 for a in accepted_bids])
            results.append(profit)
        return results


# ================================================================
# 5. Regulator / Policy Maker
# ================================================================

class Regulator:
    """Evaluates system-wide welfare, emission, and reliability."""
    def __init__(self, env):
        self.env = env
        self.controls = {"c_minus": env.params["c_minus"],
                         "c_plus": env.params["c_plus"],
```

```
                            "tau": env.params.get("tau", np.zeros(24)),
                            "R_target": env.params.get("R_target", 0.0)}

    def set_controls(self, controls):
        for k, v in controls.items():
            self.controls[k] = v
        self.env.params.update(controls)

    def social_welfare(self, demand, prices, quantities):
        """Compute social welfare = utility - total payments."""
        U = self.env.params["U_func"]
        total_payment = np.sum([p * q for p, q in zip(prices, quantities)])
        return np.sum([U(d) for d in demand]) - total_payment

    def emissions(self, quantities):
        """Compute total emissions from conventional producers."""
        alphas = self.env.params["alpha"]
        return np.sum([alphas[j] * quantities[j+1] for j in range(3)])  # j+1 skip renewable

    def reliability(self, demand, quantities):
        """Compute unmet demand penalty."""
        total_supply = np.sum(quantities)
        return np.sum(np.maximum(demand - total_supply, 0))


# ===============================================================
# 6. Unified Market Simulation Interface
# ===============================================================

def simulate_market(control_dict, uncertainty_dict):
    """
    Simulate unified market dynamics.

    Inputs:
        control_dict: dictionary of control variables for each perspective
            {
                "renewable": {"b_r": [...], "p_r": [...]},
                "regulator": {"c_minus": val, "c_plus": val, "tau": [...], "R_target": val}
            }
        uncertainty_dict: dictionary of uncertain variable realizations
            {"D_t": [...], "g_t": [...], "b_t^j": [...], "p_t^j": [...]}

    Returns:
        dict of objective values for each perspective
    """
    # ------------------ Shared environment setup ------------------
    params = {
        "mu_D": 1000, "sigma_D": 100,
        "mu_b": [300, 400, 500],
        "sigma_b": [50, 50, 50],
        "mu_p": [40, 45, 50],
        "sigma_p": [5, 5, 5],
        "c_minus": 20, "c_plus": 10,
        "g_bounds": (0, 800),
        "alpha": [0.5, 0.6, 0.8],  # emission factors
        "U_func": lambda d: 100 * np.sqrt(d)
    }

    env = MarketEnvironment(params, uncertainty_dict)

    # ------------------ Initialize modules ------------------
    renewable = RenewableProducer(env)
    renewable.set_controls(control_dict.get("renewable", {}))

    regulator = Regulator(env)
    regulator.set_controls(control_dict.get("regulator", {}))

    operator = SystemOperator(env)
    conventional = ConventionalProducers(env)

    # ------------------ Simulation across hours ------------------
    profits_r, imbalances_r = 0.0, 0.0
    total_emissions, total_welfare, reliability_penalty = 0.0, 0.0, 0.0

    for t in range(24):
```

```
        D_t = env.get_demand(t)
        gen_t = env.get_generation(t)
        conv_bids = env.get_conventional_bids(t)
        bid_r = (renewable.controls["b_r"][t], renewable.controls["p_r"][t])
        bids_all = [bid_r] + conv_bids

        # Merit-order clearing
        pi_t, q_t_all = operator.clear_market(bids_all, D_t)

        # Evaluate renewable outcomes
        profits_r += renewable.profit_objective([pi_t]*24, [q_t_all[0]]*24, [gen_t]*24)
        imbalances_r += renewable.imbalance_objective([gen_t]*24)

        # Evaluate regulator objectives
        total_welfare += regulator.social_welfare([D_t], [pi_t], q_t_all)
        total_emissions += regulator.emissions(q_t_all)
        reliability_penalty += regulator.reliability(D_t, q_t_all)

    # ------------------ Return all objectives ------------------
    results = {
        "RenewableProducer": {
            "ExpectedProfit": profits_r,
            "ImbalanceExposure": imbalances_r
        },
        "ConventionalProducers": {
            "ExpectedProfits": conventional.expected_profit_metrics([pi_t], q_t_all)
        },
        "Regulator": {
            "SocialWelfare": total_welfare,
            "Emissions": total_emissions,
            "ReliabilityPenalty": reliability_penalty
        }
    }

    return results
```

✅ Usage (conceptual example)

```python
control_inputs = {
    "renewable": {"b_r": np.full(24, 200), "p_r": np.full(24, 45)},
    "regulator": {"c_minus": 25, "c_plus": 15, "tau": np.full(24, 5.0)}
}

uncertainty_inputs = {
    "D_t": np.random.normal(1000, 100, 24),
    "g_t": np.random.uniform(100, 700, 24)
}

outputs = simulate_market(control_inputs, uncertainty_inputs)
print(outputs)
```

(You don't need to execute this — it's a complete modular structure ready for simulation or sensitivity analysis.)

🧠 Design Highlights

- **Shared environment**: `MarketEnvironment` encapsulates all global parameters and uncertainties.
- **Perspective modularity**:
  - `RenewableProducer`: control over bids, profit and imbalance objectives.
  - `ConventionalProducers`: stochastic participation, evaluated by metrics only.

- `SystemOperator`: enforces market clearing.
- `Regulator`: evaluates social welfare, emissions, reliability; can adjust policy controls.

- **Interface function `simulate_market()`:**
  - Accepts all controls and uncertain inputs.
  - Simulates the system dynamics hour by hour.
  - Returns all objective function values as a structured dictionary.

This implementation directly corresponds to the unified model specification — modular, internally consistent, and ready for extension into equilibrium or policy simulations.

> Your Python implementation is incorrect. For each hour t, you evaluate the profit outcome for 24 hours. Please refine the implementation.

You're absolutely right — in my previous code, the renewable producer's hourly profit was incorrectly aggregated over all 24 hours for every hour *t*. Let's fix that by ensuring that **each hour's outcome is computed only once per loop iteration** and then accumulated correctly.

Here's the corrected and fully consistent modular Python implementation:

```python
import numpy as np

# ===============================================================
# 1. Shared Environment
# ===============================================================

class MarketEnvironment:
    """Shared environment including parameters and uncertainties."""
    def __init__(self, params, uncertainties):
        self.params = params
        self.uncertainties = uncertainties
        self.num_hours = 24

    def get_demand(self, t):
        mu, sigma = self.params["mu_D"], self.params["sigma_D"]
        return (self.uncertainties.get("D_t", [None]*24)[t]
                or np.random.normal(mu, sigma))

    def get_conventional_bids(self, t):
        bids = []
        for j in range(3):
            mu_b, sig_b = self.params["mu_b"][j], self.params["sigma_b"][j]
            mu_p, sig_p = self.params["mu_p"][j], self.params["sigma_p"][j]
            b = (self.uncertainties.get("b_t^j", [[None]*3]*24)[t][j]
                 if "b_t^j" in self.uncertainties else np.random.normal(mu_b, sig_b))
            p = (self.uncertainties.get("p_t^j", [[None]*3]*24)[t][j]
                 if "p_t^j" in self.uncertainties else np.random.normal(mu_p, sig_p))
            bids.append((b, p))
        return bids

    def get_generation(self, t):
        g_min, g_max = self.params["g_bounds"]
        return (self.uncertainties.get("g_t", [None]*24)[t]
                or np.random.uniform(g_min, g_max))


# ===============================================================
```

```python
# 2. System Operator (Market Clearing)
# ================================================================

class SystemOperator:
    def clear_market(self, bids, demand):
        """Merit-order clearing."""
        bids_sorted = sorted(enumerate(bids), key=lambda x: x[1][1])
        supply = 0.0
        clearing_price = None
        accepted = [0.0] * len(bids)

        for idx, (b, p) in [(i, bid) for i, bid in bids_sorted]:
            if supply + b < demand:
                supply += b
                accepted[idx] = b
            else:
                accepted[idx] = max(0, demand - supply)
                clearing_price = p
                break

        if clearing_price is None:  # all bids accepted
            clearing_price = max(p for (_, p) in bids)

        return clearing_price, accepted


# ================================================================
# 3. Renewable Producer
# ================================================================

class RenewableProducer:
    def __init__(self, env):
        self.env = env
        self.controls = {"b_r": np.zeros(24), "p_r": np.zeros(24)}

    def set_controls(self, controls):
        self.controls["b_r"] = np.array(controls.get("b_r", np.zeros(24)))
        self.controls["p_r"] = np.array(controls.get("p_r", np.zeros(24)))

    def hourly_profit(self, t, price, accepted, generation):
        """Profit in hour t only."""
        c_minus, c_plus = self.env.params["c_minus"], self.env.params["c_plus"]
        tau = self.env.params.get("tau", np.zeros(24))[t]
        b_t, p_t = self.controls["b_r"][t], self.controls["p_r"][t]
        a_t = 1.0 if accepted > 0 else 0.0
        g_t = generation
        return a_t * ((p_t + tau) * b_t
                      - c_minus * max(b_t - g_t, 0)
                      - c_plus * max(g_t - b_t, 0))

    def hourly_imbalance(self, t, generation):
        b_t = self.controls["b_r"][t]
        return abs(b_t - generation)


# ================================================================
# 4. Conventional Producers
# ================================================================

class ConventionalProducers:
    def __init__(self, env):
        self.env = env

    def expected_profit_metrics(self, clearing_price, accepted):
        results = []
        for j in range(3):
            mu_p = self.env.params["mu_p"][j]
            mu_b = self.env.params["mu_b"][j]
            Cj = self.env.params.get("cost_j", [0, 0, 0])[j]
            profit = (mu_p - Cj) * mu_b * (1 if accepted[j+1] > 0 else 0)
            results.append(profit)
        return results


# ================================================================
```

```
# 5. Regulator / Policy Maker
# ==============================================================

class Regulator:
    def __init__(self, env):
        self.env = env
        self.controls = {"c_minus": env.params["c_minus"],
                         "c_plus": env.params["c_plus"],
                         "tau": env.params.get("tau", np.zeros(24)),
                         "R_target": env.params.get("R_target", 0.0)}

    def set_controls(self, controls):
        for k, v in controls.items():
            self.controls[k] = v
        self.env.params.update(controls)

    def welfare(self, demand, price, quantities):
        U = self.env.params["U_func"]
        total_payment = np.sum([price * q for q in quantities])
        return U(demand) - total_payment

    def emissions(self, quantities):
        alphas = self.env.params["alpha"]
        return np.sum([alphas[j] * quantities[j+1] for j in range(3)])  # skip renewable

    def reliability(self, demand, quantities):
        total_supply = np.sum(quantities)
        return max(demand - total_supply, 0)


# ==============================================================
# 6. Unified Market Simulation Interface
# ==============================================================

def simulate_market(control_dict, uncertainty_dict):
    """
    Simulate the unified market dynamics for 24 hours.

    Inputs:
        control_dict: dict of controls per perspective
        uncertainty_dict: dict of uncertain variables or samples

    Returns:
        dict of objective values for each perspective
    """
    params = {
        "mu_D": 1000, "sigma_D": 100,
        "mu_b": [300, 400, 500],
        "sigma_b": [50, 50, 50],
        "mu_p": [40, 45, 50],
        "sigma_p": [5, 5, 5],
        "c_minus": 20, "c_plus": 10,
        "g_bounds": (0, 800),
        "alpha": [0.5, 0.6, 0.8],
        "U_func": lambda d: 100 * np.sqrt(d)
    }

    env = MarketEnvironment(params, uncertainty_dict)
    renewable = RenewableProducer(env)
    renewable.set_controls(control_dict.get("renewable", {}))

    regulator = Regulator(env)
    regulator.set_controls(control_dict.get("regulator", {}))

    operator = SystemOperator()
    conventional = ConventionalProducers(env)

    # Initialize accumulators
    total_profit_r = 0.0
    total_imbalance_r = 0.0
    total_welfare = 0.0
    total_emissions = 0.0
    total_reliability = 0.0

    # Hourly simulation
```

```
    for t in range(24):
        D_t = env.get_demand(t)
        g_t = env.get_generation(t)
        conv_bids = env.get_conventional_bids(t)
        bid_r = (renewable.controls["b_r"][t], renewable.controls["p_r"][t])
        bids_all = [bid_r] + conv_bids

        pi_t, q_t_all = operator.clear_market(bids_all, D_t)

        # --- Renewable outcomes ---
        profit_t = renewable.hourly_profit(t, pi_t, q_t_all[0], g_t)
        imbalance_t = renewable.hourly_imbalance(t, g_t)
        total_profit_r += profit_t
        total_imbalance_r += imbalance_t

        # --- Regulator outcomes ---
        total_welfare += regulator.welfare(D_t, pi_t, q_t_all)
        total_emissions += regulator.emissions(q_t_all)
        total_reliability += regulator.reliability(D_t, q_t_all)

    results = {
        "RenewableProducer": {
            "ExpectedProfit": total_profit_r,
            "ImbalanceExposure": total_imbalance_r
        },
        "ConventionalProducers": {
            "ExpectedProfits": conventional.expected_profit_metrics(pi_t, q_t_all)
        },
        "Regulator": {
            "SocialWelfare": total_welfare,
            "Emissions": total_emissions,
            "ReliabilityPenalty": total_reliability
        }
    }

    return results
```

✅ Fixes and Improvements

- Each hour's **profit and imbalance** are now evaluated **only once per t**, not across all 24 hours.
- The renewable producer's and regulator's objectives correctly **accumulate hourly results**.
- The function **`simulate_market()`** remains the single interface, producing a structured dictionary of objective values.
- The model structure remains **modular** and **internally consistent**, suitable for extension (e.g., Monte Carlo simulation or optimization).