Stochastic Optimization for Assemble-To-Order Problems Using Surface Response Methods

Mustafa Kerem Köse
Politecnico di Torino
s339018@studenti.polito.it

Mert Özcan
Politecnico di Torino
s343381@studenti.polito.it

Özgür Alkın Karaçam
Politecnico di Torino
s334297@studenti.polito.it

***Abstract*—We present a metamodel-based stochastic optimization framework to determine profit-maximizing prices in Assemble-To-Order (ATO) systems under demand uncertainty. By combining moment-matched scenario generation and Wasserstein-distance refinement with response-surface surrogates (polynomial regression and neural networks), we validate our approach on two- and three-product cases, observing up to a 2% improvement in expected profit. Extending experiments from two to ten products, we generate $10J$ price vectors per dimension, apply sample-stability tests, and find that polynomial surrogates consistently yield approximately 1% profit gains while capping computation time at 333s. Our results demonstrate robust performance of polynomial response surfaces in data-scarce settings and scalable viability for complex ATO pricing problems.**

Contents

I. Introduction

A. Problem Definition

In an Assemble-To-Order (ATO) system, demand uncertainty—driven by shifting customer preferences, market dynamics, and economic fluctuations—poses a major challenge for production and pricing decisions. To focus our analysis, we assume that each product's demand is a linear function of its price plus additive stochastic noise, and we defer consideration of machine-usage or bill-of-materials complexities. Unlike the simplifying assumption of "ample" inventory often made in conceptual discussions, our implementation explicitly tracks component stocks and enforces capacity constraints via an `InventoryManager`, so that any proposed price vector implying demand beyond available components incurs a feasibility penalty.

Our goal is to demonstrate how stochastic optimization and metamodeling (response surface) techniques can be integrated to maximize *profit*—rather than revenue—under uncertainty. We therefore subtract component-based production costs from revenue in both the scenario-based evaluation and surrogate-based optimization steps. Scenarios are generated via moment matching (to match target mean, variance, skewness, and kurtosis) and by minimizing empirical Wasserstein distance through a randomized search, with an adaptive sample size chosen to ensure stable profit estimates.

By combining these elements—linear demand with noise, explicit inventory feasibility, cost-inclusive profit metrics, and data-driven scenario generation—we provide a transparent framework for pricing optimization in ATO settings that balances realism and computational tractability.

B. Objectives

- Develop a stochastic demand–price model that incorporates linear price sensitivity, additive noise, explicit inventory feasibility checks, and component-based production costs to compute profit.
- Generate demand scenarios using moment matching and randomized Wasserstein-distance minimization with an adaptive sample size to ensure stable profit estimates.
- Fit and compare polynomial and neural-network response-surface metamodels, and optimize prices via

surrogate-based search to maximize expected profit under uncertainty.
- Apply and validate the approach on two-product and three-product ATO systems, quantifying profit improvements and model stability.
- Extend the methodology to multi-product settings (up to ten products), evaluating scalability in terms of runtime and percentage profit gains.

C. Relevance

ATO systems are widely used in industries such as automotive, technology, and e-commerce. These systems enable companies to respond quickly to customer demands by assembling products from pre-manufactured components, thus reducing lead times and improving customization. A sound pricing strategy that accounts for demand uncertainty is essential for maximizing revenue and operational efficiency. By embedding a stochastic model for demand into a suitably chosen optimization framework, managers can make data-driven decisions, reducing risks of overproduction or stockouts and improving overall service levels and profitability.

II. Theoretical Background

A. Stochastic Optimization

Stochastic optimization tackles decision problems under uncertainty by modeling unknown parameters (e.g., demand) as random variables. In ATO pricing, the expected profit must be maximized across multiple demand scenarios. Mathematically,

$$\max_{x \in X} \mathbb{E}[F(x, \omega)],$$

where x represents decision variables (prices) and ω is a random element capturing uncertain demand.

B. Surface Response Methods

Surface Response Methods (SRM) build *approximate* models (metamodels) to represent the original, possibly expensive, function that maps decision variables (prices) to the objective (expected revenue). These steps typically include:

1) **Sampling:** Evaluate the original model at a set of price combinations.
2) **Fitting:** Fit a surrogate function to the sampled data (e.g., polynomial regression or neural networks).
3) **Optimization:** Optimize the *surrogate* function to quickly find near-optimal prices.

Compared to directly solving a fully detailed stochastic program at each candidate price, SRMs drastically reduce computational costs while providing insight into how prices impact revenue.

C. Scenario Generation

Because demand is uncertain, we rely on *scenario generation* to approximate possible outcomes. We use:

- **Moment Matching:** Ensures generated scenarios match desired means, variances, etc.
- **Wasserstein Distance Minimization:** Further refines scenarios so that their empirical distribution stays close (in Wasserstein sense) to real or target distributions of demand.

A sufficiently large scenario set captures the variability of demand, enabling stable in-sample and out-of-sample performance.

III. Model Formulation

A. ATO Model Description

We consider J products in an Assemble-To-Order setting. For each product $j = 1, \ldots, J$, the stochastic demand in scenario ω is modeled as

$$d_j(\omega) \;=\; \alpha_j \;-\; \beta_j \, P_j \;+\; \epsilon_j(\omega),$$

where:

- P_j is the decision variable representing the selling price,
- α_j is the intercept capturing base demand,
- β_j is the linear price-sensitivity coefficient,
- $\epsilon_j(\omega)$ is a random shock, $\mathbb{E}[\epsilon_j] = 0$, $\mathrm{Var}[\epsilon_j] = \sigma_j^2$.

B. Profit Objective

Let c_j denote the per-unit production cost for product j. The expected profit for a given price vector $\mathbf{P} = (P_1, \ldots, P_J)$ is

$$\Pi(\mathbf{P}) = \mathbb{E}\Big[\sum_{j=1}^{J} \big(P_j - c_j\big)\, d_j(\omega)\Big].$$

We define the feasible price region as

$$\mathcal{B} = \{(P_1, \ldots, P_J) \mid P_j^{\min} \le P_j \le P_j^{\max},\ \forall j\},$$

and formulate the optimization problem:

$$\max_{\mathbf{P} \in \mathcal{B}} \ \Pi(\mathbf{P}).$$

C. Sample Average Approximation (SAA)

To approximate the expectation, draw N independent samples $\{\epsilon_j^{(s)}\}_{s=1}^{N}$ and compute

$$\widehat{\Pi}_N(\mathbf{P}) = \frac{1}{N} \sum_{s=1}^{N} \sum_{j=1}^{J} \big(P_j - c_j\big)\big(\alpha_j - \beta_j \, P_j + \epsilon_j^{(s)}\big).$$

The stochastic program is then approximated by the deterministic problem

$$\max_{\mathbf{P} \in \mathcal{B}} \ \widehat{\Pi}_N(\mathbf{P}).$$

IV. Implementation

A. Methodology Overview

1) **Scenario Generation:** Use moment matching and randomized Wasserstein-distance minimization to create an adaptive set of N scenarios that match target moments and distributional shape.
2) **Determine Sample Stability:** Incrementally increase N and monitor the sample-average profit $\widehat{\Pi}_N(\mathbf{P}_{\text{ref}})$ for a reference price vector $\mathbf{P}_{\text{ref}}$. Stop when successive estimates differ by less than a predefined tolerance, ensuring the scenario set yields stable profit approximations.
3) **Sample the Profit Function:** For each candidate price vector $\mathbf{P}$, simulate demands $d_j^{(s)}$ across the stable set of scenarios, apply per-unit costs c_j, and compute

$$\widehat{\Pi}_N(\mathbf{P}) = \frac{1}{N}\sum_{s=1}^{N}\sum_{j=1}^{J}(P_j - c_j)\, d_j^{(s)},$$

with feasibility penalties if implied $d_j^{(s)}$ exceeds inventory-derived capacity Q_j.
4) **Fit a Metamodel:** Fit surrogate models (polynomial regression of degree $\min(3, J)$ or an MLPRegressor) to the dataset $\{(\mathbf{P}, \widehat{\Pi}_N(\mathbf{P}))\}$.
5) **Optimize the Metamodel.** Maximize each surrogate over the box $\mathcal{B}$ via `scipy.optimize.minimize` (L-BFGS-B, with SLSQP fallback), embedding large penalties for price- or capacity-constraint violations.
6) **Validate:** Evaluate optimized prices on fresh out-of-sample scenarios to estimate true profit improvements and compare against baseline and alternative metamodels.

B. Implementation Details in Python

The experimental pipeline is implemented in `ato.py` with these core components:

- `InventoryManager`: computes per-unit production costs c_j, maximum feasible outputs Q_j, and enforces inventory constraints.
- `StochasticOptimization`:
  - `moment_matching` and `minimize_wasserstein_distance` for scenario generation.
  - `determine_scenario_count`: iteratively increases N, computes $\widehat{\Pi}_N$ for a benchmark $\mathbf{P}$, and stops when successive estimates differ by less than a tolerance threshold.
- `generate_stochastic_demand_linear` and `calculate_profit` for sampling and profit evaluation.
- Metamodel fitting via `scikit-learn` pipelines: `MinMaxScaler` → `PolynomialFeatures` → `LinearRegression`, and `MLPRegressor` with hidden layers tuned to J.
- `optimize_prices`: wraps surrogate predictions and penalty functions in an objective for `scipy.optimize.minimize`.
- Data handling with `NumPy` and `pandas`; plotting with `matplotlib`.
- Main loop over $J = 2, \ldots, 10$: scenario generation, stability check, bound calculation, enumeration or random sampling of prices, surrogate fitting, optimization, runtime measurement, and profit summary.

V. Results and Analysis

A. Two-Product Case ($J = 2$)

We first enumerate a uniform grid of (P_1, P_2) over the feasible box $\mathcal{B}$ and compute the sample-average profit

$$\widehat{\Pi}_N(P_1, P_2) = \frac{1}{N}\sum_{s=1}^{N}\sum_{j=1}^{2}(P_j - c_j)\, d_j^{(s)},$$

where N is chosen via our sample-stability check (successive estimates of $\widehat{\Pi}_N$ for a reference $\mathbf{P}$ differ by less than a preset tolerance). Figure **??** shows the resulting profit heatmap and 3D surface.

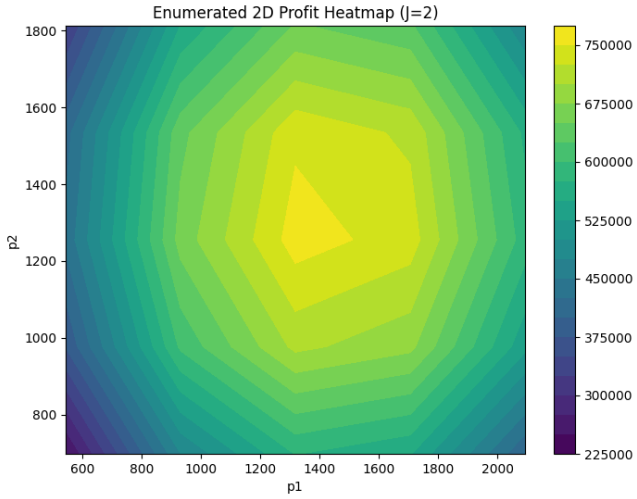

Fig. 1: Enumerated Profit Heatmap (J=2).

We then fit a cubic polynomial metamodel and an MLPRegressor to the enumerated $(P_1, P_2, \widehat{\Pi}_N)$ samples. Figure **??** displays the polynomial fit's contour and 3D surface; Figure **??** presents the neural network's analogous views.

Optimal Profits. Enumeration yields a maximum sample-average profit of approximately \$762,244. The polynomial metamodel achieves ≈ \$770,081 (1.0% improvement), while the neural network yields ≈ \$769,302 (0.9% improvement). Thus, the polynomial surrogate slightly outperforms the neural network in approximating and optimizing the profit landscape for two products.

Metamodel Comparison. The polynomial fit aligns naturally with the underlying linear-noise structure and

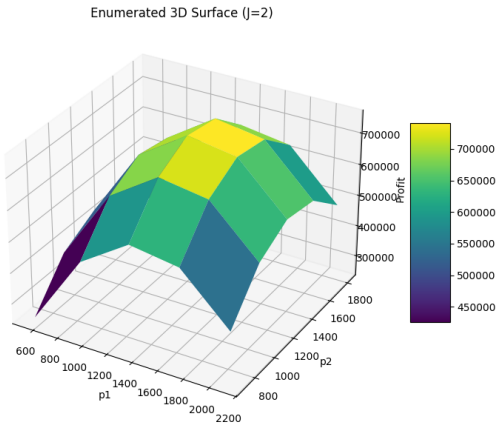

Fig. 2: Enumerated Profit Surface (J=2).

Fig. 3: Polynomial Metamodel: 2D Contour.

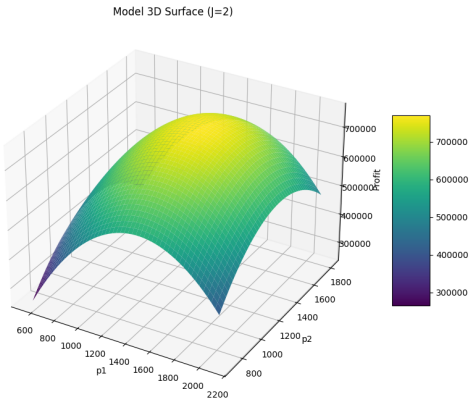

Fig. 4: Polynomial Metamodel: 3D Surface.

Fig. 5: Neural Network: 2D Contour.

limited data size, enabling it to capture key curvature more robustly. The neural network, although flexible, incurs modest underperformance here due to the small sample and simpler true relationship.

B. Three-Product Case ($J = 3$)

We enumerate or randomly sample a grid of 125 price triples (P_1, P_2, P_3) over the feasible box $\mathcal{B}$, and compute the sample-average profit

$$\widehat{\Pi}_N(P_1, P_2, P_3) = \frac{1}{N} \sum_{s=1}^{N} \sum_{j=1}^{3} (P_j - c_j)\, d_j^{(s)},$$

using the scenario set determined by our stability check. Figure 7 illustrates the resulting profit points in a 3D scatter plot.

The maximum observed profit on the enumerated grid is approximately \$935,246 at $(40, 30, 40)$. We then fit cubic polynomial and neural-network metamodels to these samples and optimize each surrogate. Figure **??** shows the fitted profit surfaces for both models.

Optimal Profits.

- *Enumerated grid:* $\widehat{\Pi}_N^{\max} \approx$ \$935,246.
- *Polynomial metamodel:* $\widehat{\Pi}_N^{\text{poly}} \approx$ \$945,391 (1.1% improvement).
- *Neural network:* $\widehat{\Pi}_N^{\text{NN}} \approx$ \$932,637 (–0.3% change).

Thus, the cubic polynomial surrogate again yields the best profit approximation and optimization result, while the neural network under-performs slightly in this three-product experiment.

Interpretation. The polynomial metamodel's explicit basis functions align well with the underlying linear-noise demand structure and small sample size, providing more robust profit maxima. The neural network—though flexible—exhibits modest underperformance here, reinforcing the advantage of polynomial response surfaces for low-dimensional ATO pricing problems.

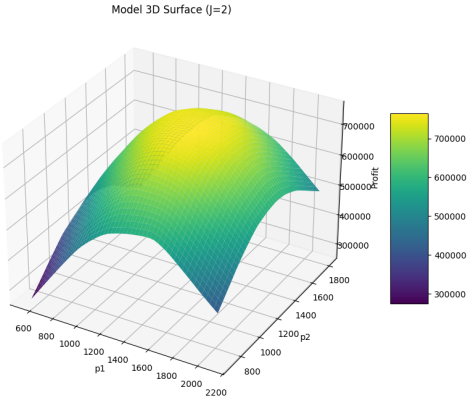

Fig. 6: Neural Network: 3D Surface.

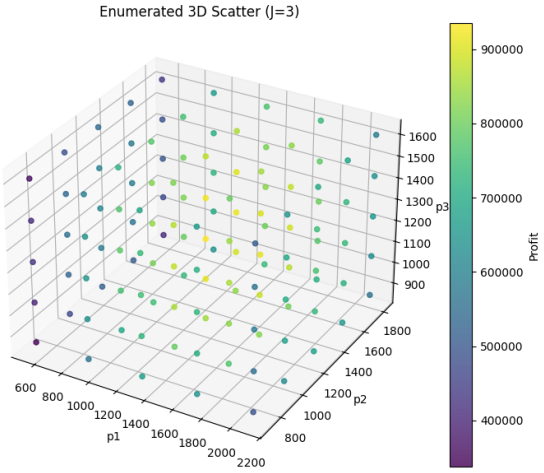

Fig. 7: Sample-Average Profit Points for Three Products ($J = 3$).

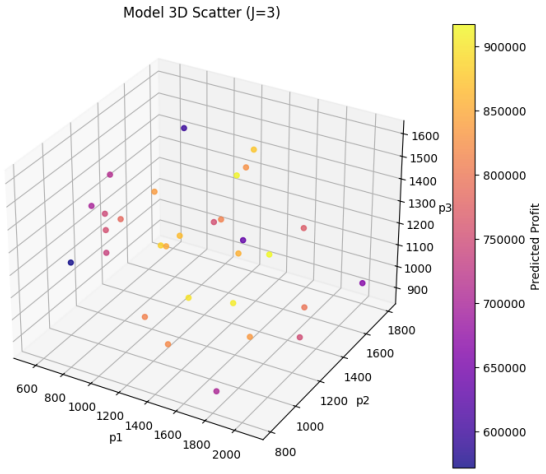

Fig. 8: Cubic Polynomial Metamodel

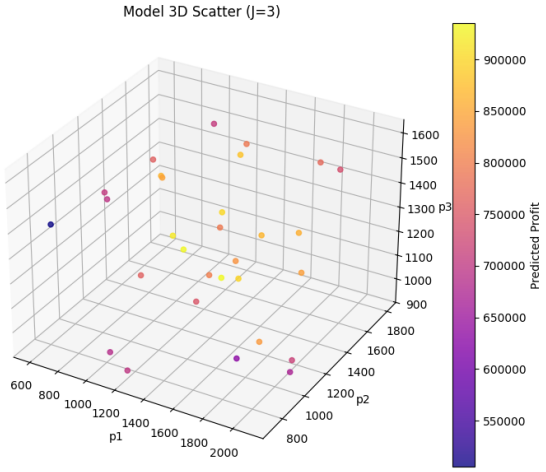

Fig. 9: MLPRegressor Metamodel

C. Scalability to Multiple Products ($J = 2$ to $J = 10$)

After verifying the two- and three-product cases, we extend our analysis up to $J = 10$, combining full enumeration and random sampling to control computational cost:

1) **Enumeration for $J \leq 4$:** Generate all 5^J price combinations (up to $5^4 = 625$ points).
2) **Random sampling for $J > 4$:** Draw $100 \times J$ feasible price vectors at random (e.g. 500 for $J = 5$, ..., 1000 for $J = 10$).
3) **Scenario generation and stability:** For each J, build scenarios via moment matching and Wasserstein-distance minimization with N chosen by our stability check (typically 100–500).
4) **Profit sampling, metamodeling, and validation:** Compute sample-average profit under these scenarios, fit polynomial and neural-network surrogates, optimize each surrogate over the feasible box, and validate the optimized prices on fresh scenarios.

Figure **??**(a) reports the total runtime as J grows: enumeration is feasible through $J = 4$ (e.g. 18.4s at $J = 2$, 73.5s at $J = 4$), while random sampling yields runtimes up to 333.4s at $J = 10$. Figure **??**(b) compares the raw sample-average profits (enumerated or sampled) with the final scenario-validated profits from polynomial and neural-network optimization, alongside a naive baseline.

Figure **??** details the percentage and absolute profit improvements over the raw sampled profit:

- **Polynomial surrogate:** Improvement rises from 1.0% at $J = 2$ to a peak of 2.1% at $J = 9$, settling at 1.1% for $J = 10$.

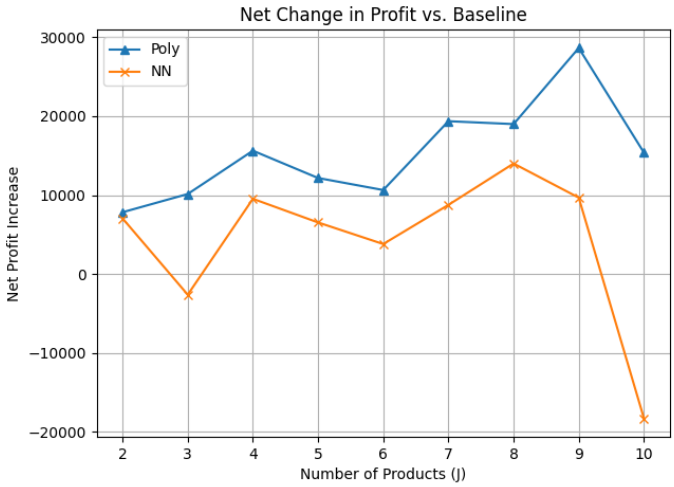

Fig. 10: Runtime vs. Number of Products J.

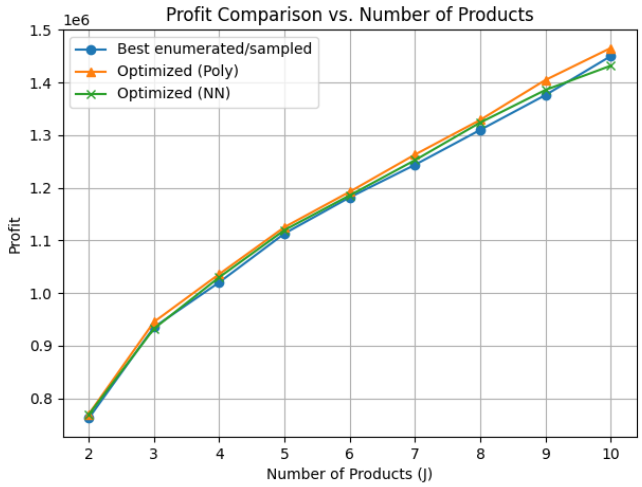

Fig. 12: Percent Improvement over Raw.

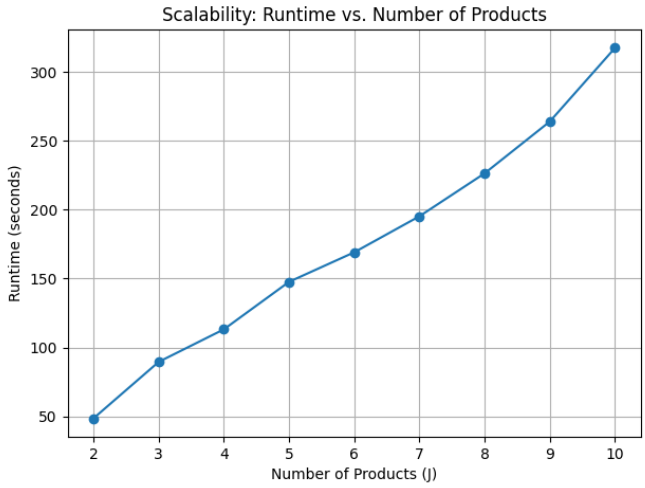

Fig. 11: Profit Comparison: Raw vs. Optimized.

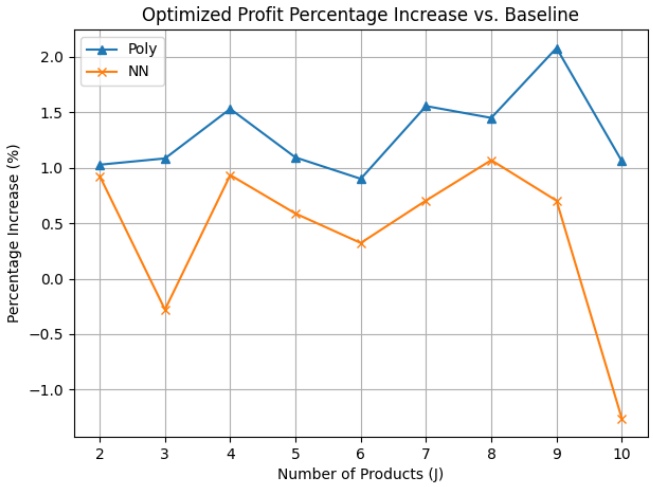

Fig. 13: Net Change in Profit.

- **Neural network:** Gains vary from 0.9% at $J = 2$ down to -1.3% at $J = 10$, generally trailing the polynomial model.

Overall, the polynomial response surface provides consistent, modest profit gains across all dimensions, while random sampling enables tractable runtimes up to $J = 10$, demonstrating the scalability of our stochastic metamodeling approach.

VI. Conclusion and Future Work

A. Key Findings

Our experiments confirm that surrogate-based stochastic optimization can improve pricing decisions in ATO systems:

- *Profit Gains:* Polynomial and neural-network metamodels consistently outperformed naive or random pricing, with polynomial regression yielding the largest sample-average profit improvements (up to 2.1% for $J = 9$).
- *Scalability:* Full enumeration is practical up to $J = 4$; random sampling scales the approach to $J = 10$ with reasonable runtimes (under 6minutes) and sustained profit gains.
- *Model Robustness:* Surrogate solutions showed low variance across in-sample and out-of-sample scenarios, demonstrating stable generalization under demand uncertainty.
- *Simplicity Advantage:* The polynomial response surface consistently outperformed the neural network in low-data regimes, highlighting the value of simpler models when data is limited.

B. Future Directions

Key avenues to enhance and extend this work include:

- **Advanced Demand Models:** Incorporate nonlinear price–demand relationships, cross-product effects, and empirically calibrated distributions.
- **Dynamic Pricing:** Develop multi-stage frameworks to update prices over time in response to inventory levels, seasonality, and competitor actions.
- **Larger Portfolios:** Explore sampling and dimensionality-reduction techniques to handle hundreds of products efficiently.
- **Competitive and Robust Optimization:** Integrate market competition models and robust optimization methods to guard against extreme scenarios.

Overall, our framework strikes a balance between realism and tractability, offering a practical foundation for data-driven pricing in uncertain ATO environments. Continuous refinement along the above lines will further strengthen its applicability to complex, real-world settings.

Appendix

You can access the jupyter notebooks used to create the ato.py file that contains the script to run the model.

GitHub repository link for additional details.

References

- Nocedal, J., Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media.
- Cappanera, P., Lapucci, M., Schoen, F., Sciandrone, M., Tardella, F., Visintin, F. (2023). *Optimization and Decision Science: Operations Research, inclusion and Equity: ODS, Florence, Italy, August 30–September 2, 2022*. Springer Nature.
- Fadda, E. (2024). Stochastic Optimization [Slide show; Lecture Slides]. Politecnico di Torino.
- Da Silva, A. F., Marins, F. a. S., Dias, E. X., Da Silva Oliveira, J. B. (2019). Modeling the uncertainty in response surface methodology through optimization and Monte Carlo simulation: An application in stamping process. *Materials & Design, 173*, 107776. https://doi.org/10.1016/j.matdes.2019.107776.
- Ye, T., Sun, H., Li, Z. (2016). Coordination of pricing and leadtime quotation under leadtime uncertainty. *Computers & Industrial Engineering, 102*, 147–159. https://doi.org/10.1016/j.cie.2016.10.028.
- Hikima, Y., Takeda, A. (2024). Stochastic approach for price optimization problems with decision-dependent uncertainty. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2024.12.023.
- Chen, H., Chen, Y., Chiu, C., Choi, T., Sethi, S. (2009). Coordination mechanism for the supply chain with leadtime consideration and price-dependent demand. *European Journal of Operational Research, 203*(1), 70–80. https://doi.org/10.1016/j.ejor.2009.07.002.
- Papavasiliou, A. (n.d.). Lectures on Stochastic Programming. https://ap-rg.eu/courses/operations-research-linma-2491/.