Hybrid Genetic Algorithm and XGBoost Framework for Price Optimization with Cross-price Elasticity

Kritsada Ruangthawee
Thai-Nichi International College
Thai-Nichi Institute of Technology
Bangkok, Thailand
ru.kritsada_st@tni.ac.th

Suwan Juntiwasarakij
Thai-Nichi International College
Thai-Nichi Institute of Technology
Bangkok, Thailand
suwan.jun@tni.ac.th

Kawiwat Amnatchotiphan
Thai-Nichi International College
Thai-Nichi Institute of Technology
Bangkok, Thailand
kawiwat@tni.ac.th

***Abstract*— Pricing in modern retail environments is an increasingly complex challenge characterized by non-linear demand patterns and intricate product interdependencies known as cross-price elasticity. Traditional pricing strategies, primarily those relying on rule-based systems or linear regression models, often fail to capture these high-dimensional behaviors, which can lead to a significant decrease in profit through cannibalization effects. This paper proposes a novel hybrid optimization framework that combines an eXtreme Gradient Boosting (XGBoost) model for demand forecasting with a Genetic Algorithm (GA) serving as the core heuristic search engine. The approach uses the XGBoost model as a fitness function to estimate base demand, which is then adjusted through a cross-price elasticity calculation to simulate realistic market interactions. To manage the immense computational load, the framework employs parallel computing via store-by-store decomposition, treating each retail location as an independent optimization task. Additionally, an adaptive mutation strategy including a triggered hypermutation mechanism is used to prevent the search from stagnating in local optima during exploration of the vast search space created by unique SKU pricing combinations. Experimental results, constrained within a 4-hour operational window, show that the Hybrid GA approach significantly outperforms Brute Force strategies and historical human-driven pricing. The findings demonstrate a substantial profit increase of up to 7% in total gross margin. This research offers a scalable, production-ready decision-support system that effectively balances portfolio-wide profitability with real-world inventory and market constraints.**

Keywords—Price Optimization, Genetic Algorithm, XGBoost, Cross-Price Elasticity, Parallel Computing

I. INTRODUCTION

In modern retail management, price optimization has become a critical driver of profitability and competitive advantage. Determining optimal prices across a large portfolio of products, however, remains a highly complex task, often formulated as a large-scale combinatorial optimization problem [1]. This complexity arises from the need to jointly model individual price–demand relationships and interdependencies among products, particularly cross-price elasticity effects, where price changes in one item influence the demand of others [2]–[4]. Consequently, the resulting search space grows exponentially with the number of stock-keeping units (SKUs), making exhaustive optimization computationally infeasible.

Conventional pricing approaches, including rule-based systems and traditional econometric approaches, are limited in their ability to capture the nonlinear and high-dimensional dynamics of real-world consumer behavior [5]. Moreover, such methods frequently evaluate products independently, overlooking portfolio-level interactions and leading to suboptimal outcomes such as cannibalization, where gains in one product diminish overall profitability [4], [6].

To address these challenges, this paper proposes a hybrid optimization framework that integrates eXtreme Gradient Boosting (XGBoost) with Genetic Algorithms (GAs). The proposed approach leverages XGBoost to model complex demand patterns using historical sales data, competitor pricing, and contextual variables. This predictive model is then embedded within a GA-based optimization process, which efficiently explores the high-dimensional pricing space through evolutionary search mechanisms.

The primary contribution of this work lies in the seamless integration of demand prediction and price optimization within a unified framework. Specifically, cross-price elasticity effects are incorporated directly into the optimization loop, enabling simultaneous evaluation of interdependent pricing decisions across a large product portfolio. The proposed method also incorporates practical business constraints, including inventory limitations and operational rules, ensuring real-world applicability.

Experimental results demonstrate that the proposed framework consistently outperforms baseline approaches, including random search and historical pricing strategies, in terms of overall profit maximization. These findings highlight the potential of hybrid machine learning and metaheuristic approaches for scalable and effective retail price optimization.

II. LITERATURE REVIEW

Retail price optimization has been extensively studied in the presence of cross-price elasticity, which introduces significant complexity due to interdependencies among substitutable and complementary products [2], [3]. Early approaches typically relied on econometric models or simplified demand functions, which often assume linear relationships and limited interactions among products. While computationally efficient, these models are generally inadequate for capturing the nonlinear and high-dimensional characteristics of modern retail environments [4], [5].

Recent advances in machine learning have significantly improved the accuracy of demand forecasting. In particular, tree-based ensemble methods such as XGBoost have demonstrated strong performance in modeling complex, nonlinear relationships in large-scale retail datasets [7]. These models effectively incorporate diverse features, including historical sales, pricing strategies, and external factors. However, despite their predictive strength, such approaches are inherently descriptive and do not directly provide optimal pricing decisions under operational constraints [1].

To address optimization challenges, metaheuristic algorithms have been widely adopted. Among these, GAs have proven effective in solving large-scale, combinatorial optimization problems due to their ability to perform global search without requiring gradient information [8]. Applications of GAs in retail contexts include pricing, assortment planning, and inventory optimization. Nevertheless, many existing studies employ GAs in conjunction with simplified or exogenously defined demand models, limiting their ability to reflect real-world demand dynamics.

A key limitation in the existing literature is the separation between demand forecasting and price optimization. Most frameworks treat these components as sequential and independent processes, which can lead to suboptimal decisions due to the lack of feedback between prediction and optimization stages. Furthermore, prior work often simplifies or neglects cross-product interactions, reducing the effectiveness of portfolio-level pricing strategies [6].

This study addresses these gaps by proposing an integrated framework that combines XGBoost-based demand prediction with GA-driven price optimization. Unlike prior approaches, the proposed method embeds the predictive model directly within the optimization loop, enabling dynamic evaluation of pricing decisions under realistic constraints. By incorporating cross-price elasticity, inventory limitations, and competitive factors into a unified system, the framework provides a scalable and practical solution for multi-product retail price optimization.

III. Methodology

This section outlines the research process and the technical framework developed to solve the multi-product price optimization problem. The system is designed to integrate high-dimensional demand forecasting with a heuristic search engine to maximize total portfolio profitability.

A. *Data Acquisition and Feature Engineering*

- Inventory and Supply Chain Logic: To account for real-world supply constraints, daily inventory limits are calculated using current stock levels and lead times.

$$Inventory_{daily} = \frac{Product\ Quantity}{Lead\ Time\ Day} \tag{1}$$

  This derived attribute ensures the model penalizes pricing strategies that generate demand exceeding physical fulfillment capabilities.

- Temporal Competitor Benchmarking: Competitor price data are aggregated into weekly averages. However, rather than using a single weekly value, the system preserves seven separate columns representing each day of the week. This structure allows the optimization engine to capture day-of-week variations in consumer behaviors and competitor pricing strategies.
- Environmental and Regional Context: Sales data is enriched with daily weather indices (temperature and rain) and a store income index. The latter serves as a scaling factor to adjust base demand predictions according to the purchasing power of specific store locations.
- The dataset utilized for this research was provided by CP Axtra Public Company Limited, comprising historical sales records, competitor pricing, and inventory levels across multiple retail locations in Thailand.

B. *Search Space and Pricing Constraints*

To ensure that the optimization remains within a commercially viable range, the search space for each SKU is strictly governed by a business rule. This approach prevents the algorithm from suggesting extreme prices that could damage brand equity or lead to unsustainable losses.

- Valid Price Genes Formulation: For every SKU, a customized set of "allowable prices" is generated. These valid genes are constrained to a ±30% range relative to the product's regular price, which suggests that significant deviations from a consumer's internal reference price can lead to asymmetric demand responses [4].
  - Upper Bound: Regular_Price×1.30
  - Lower Bound: Regular_Price×0.70
- SKU-Specific Customization: The GA does not pick from a global price pool; instead, it selects values from an SKU-by-SKU lookup table. This ensures that high-value items and budget items are optimized within their respective market segments.
- Commercial Logic: By constraining the gene search space to this ±30% interval, the framework effectively filters out sub-optimal or unrealistic candidates before they enter the evolutionary loop. This focuses the GA computational power on refining the most effective price points within an acceptable business threshold.

C. *Demand Modeling with XGBoost*

A predictive model is constructed using XGBoost to estimate the base demand (Q_{base}) for each SKU.

- Feature Selection: The model utilizes a robust feature set, including historical sales, temporal indicators, and competitor price indices
- Inference Optimization: To handle the high frequency of evaluations required by the optimization loop, the system utilizes the native booster object and in-place predict functionality. This bypasses the overhead of

repeated data matrix construction, enabling rapid fitness evaluations [8].

IV. Genetic Algorithm

In this research, a GA is utilized as the core optimization engine to navigate the high-dimensional search space of multi-product pricing [9]. Unlike traditional derivative-based methods, the GA is uniquely suited for this problem because it can handle the non-linear and discrete nature of retail price points across multiple stores and SKUs.

The algorithm operates by evolving a population of candidate pricing strategies, where each individual (chromosome) represents a complete set of prices for the store's entire portfolio. The following biologically inspired mechanisms drive the evolutionary process.

A. Configuration and Optimization Strategy

GA performance is dependent on tuning key parameters like population size, crossover rate, mutation rate, tournament size, and termination criteria. Optimal settings typically involve a balance between exploring new solutions and exploiting known good solutions to avoid a local optimum point.

- Population Initialization: The process begins with a population of n individuals. Each individual is initialized using a set of valid price genes derived from historical regular prices, ensuring the search starts within a commercially viable range.
- Selection via Tournament: To identify superior pricing strategies, a tournament selection method is employed. This mechanism selects the best-performing individuals (parents) based on their fitness scores, maintaining a balance between high-quality solutions and genetic diversity.
- Two-Point Crossover: New candidate solutions (children) are generated through a two-point crossover operator. This allows the algorithm to combine successful pricing segments from different parents, effectively exploring new combinations of product prices that might yield higher aggregate profits.
- Adaptive Mutation: To avoid local optima [10], the framework utilizes an adaptive mutation strategy. The mutation rate begins at 0.2 and decays for fine-tuning. If the population's best fitness stagnates for 500 generations, a hypermutation trigger increases the rate to 0.4. This injection of genetic diversity forces the search into unexplored price regions, effectively bypassing premature convergence.
- Termination Criteria: The 4-hour timeout is strategically selected to align with retail operational windows while managing the combinatorial complexity of price combinations. Compared to Simulated Annealing, Particle Swarm Optimization, or Bayesian Optimization, the proposed GA provides superior convergence in high-dimensional spaces by leveraging parallel evaluation and adaptive mutation, ensuring robust global search efficiency where traditional point-based or swarm-based methods typically stagnate [11], [12].

B. Fitness Function

The fitness function represents the most critical component of the GA framework. The quality of the final pricing strategy is directly dependent on the accuracy and mathematical integrity of this function. In this research, the fitness function is designed to evaluate the total portfolio profitability while accounting for complex market dynamics and operational constraints.

Furthermore, the fitness function is the most computationally intensive part of the evolutionary process, as it must be executed for every individual in the population across thousands of generations. To address this, the framework implements several high-performance computing strategies:

- Hybrid Evaluation Logic: Each fitness calculation integrates the following three distinct layers.

  1. Demand Forecasting: Utilize the XGBoost model to predict base consumer demand (defined as $Q_{i,s}^{base}$).

  2. Interaction Simulation: Apply the cross-price elasticity to calculate cannibalization and substitution effects.

  3. Constraint Validation: Check the predicted demand against daily inventory limits to ensure logistical feasibility (defined as $Q_{i,s}^{final}$).

$$Q_{i,s}^{final} = Q_{i,s}^{base} \times \prod_{j \in J_i} \left[1 - E_{i,j} \left(\frac{P_j - P_{j,s}^{reg}}{P_{j,s}^{reg}} \right) \right] \quad (2)$$

where $Q_{i,s}^{final}$ is the quantity of item i at store s, $Q_{i,s}^{base}$ is the original quantity before cross-elasticity, J_i is the set of all items that have a cross-elasticity relationship with item i, $E_{i,j}$ is the cross-elasticity coefficient between item i and item j, P_j is the newly proposed price of the related item j, and $P_{j,s}^{reg}$ is the regular price of item j at store s.

C. Flowchart

Fig. 1 illustrates the flowchart of the proposed optimization framework, from data ingestion to final price selection. The flowchart shows the GA optimization process with parallel computing, where the XGBoost demand predictor serves as the fitness function.

A key architectural feature of the proposed system is the use of parallel computing through store-by-store decomposition. Because retail price optimization for a large network is computationally intensive, the framework distributes the workload by treating each store as an independent optimization task. This design enables multiple instances of the GA to run simultaneously across different processing cores. As a result, the system can efficiently scale to hundreds of store locations while ensuring that the full optimization process is completed within the practical 4-hour operational window.

The logic flow for each parallel thread is depicted in Fig. 1 and follows a structured sequence:

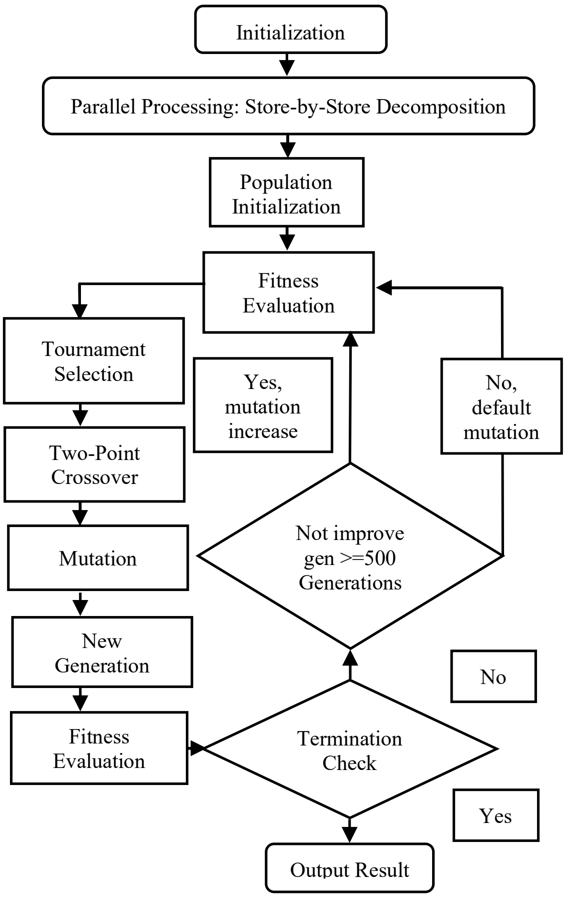

Fig. 1. GA flowchart with parallel computing.

1. Data Initialization: Ingest environmental and economic datasets, including cross-price elasticity, weather indices, and historical sales.
2. Parallel Processing: Distribute the workload via store-by-store decomposition across multiple CPU cores to ensure efficiency within a 4-hour window.
3. Population Initialization: Generate initial candidate pricing strategies constrained within a ±30% regular price window.
4. Fitness Evaluation: Use the XGBoost predictor and cross-price elasticity logic to calculate baseline portfolio profitability.
5. Selection and Genetic Operations:
    a. Tournament Selection: Select top-performing individuals to pass genetic traits to the next generation.
    b. Two-Point Crossover: Recombine parent pairs to produce diverse offspring with varied pricing combinations.
    c. Adaptive Mutation: Apply random alterations to prevent premature convergence into local optima.
6. Iterative Re-evaluation: Update the population ranking by calculating fitness scores for the new offspring.
7. Termination Check: Trigger hypermutation if fitness stagnates for 500 generations, or export the optimal strategy if criteria are met.

D. Adaptive Mutation Strategy for Escaping Local Optimum Point.

A significant challenge in evolutionary optimization is the tendency of the population to converge prematurely on a local optimum, where the algorithm ceases to find better solutions despite further iterations. To mitigate this, the framework implements an adaptive mutation strategy.

The system utilizes a dual-phase approach:

- Scheduled Decay: The mutation rate starts at a baseline of 0.2 and gradually decreases over time, allowing for the fine-tuning of solutions as the search progresses.
- Triggered Hypermutation: The system continuously monitors the improvement of the best fitness score across generations. If the best fitness score fails to improve for 500 consecutive generations, the algorithm identifies potential stagnation. In response, the mutation rate is dynamically spiked to 0.4

This temporary surge in genetic volatility forces the search into previously unexplored regions of the pricing landscape by injecting high diversity into the population. Once a new improvement in fitness is detected, the mutation rate resets to its baseline decay, allowing the algorithm to resume the balance between exploration (searching new areas) and exploitation (refining known good solutions). This mechanism ensures the optimization remains robust and capable of escaping local traps.

E. Balancing Exploration and Exploitation.

A robust optimization process relies on the delicate balance between two competing mechanisms: exploration and exploitation.

- Exploration involves searching new and unvisited areas of the search space. The aim is to discover new, potentially better solutions by broadening the search horizon. Exploration helps in avoiding local optima by introducing diversity in the search process.
- Exploitation focuses on refining and improving the current solution by searching its immediate neighborhood. The aim is to optimize known good solutions and converge to the best possible solution within the local vicinity.

In this framework, we implement a phase-based mutation strategy to manage this balance effectively. Initially, the algorithm utilizes a low mutation rate to prioritize exploitation, allowing the GA to fine-tune the pricing candidates and quickly converge on a strong solution.

However, when the search identifies a stagnation in a local optimum (defined as no improvement for 500 generations), the system automatically triggers an increased mutation rate. This transition shifts the algorithm's focus toward exploration, introducing the genetic volatility necessary to escape the local optima and discover new, more profitable regions of the search space.

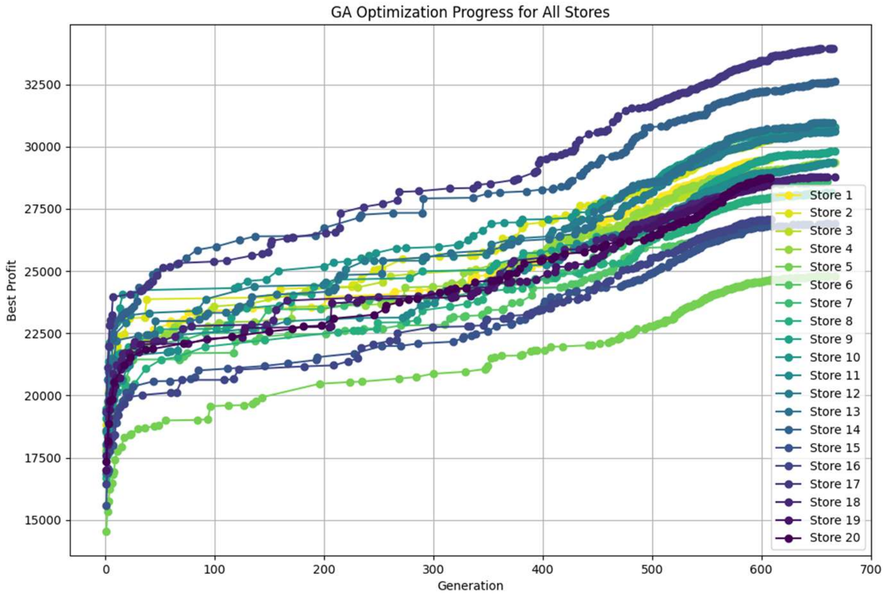

Fig. 2. Hybrid GA.

V. Results and Discussion

A. Comparative Analysis: Hybrid GA vs. Baselines and Alternative Optimizers

To validate the efficacy of the proposed framework, the hybrid GA-XGBoost model was benchmarked against two primary baselines: Brute Force (Random Search) and Human Selection (Historical Pricing). This three-way comparison demonstrates the model's ability to outperform both stochastic computational methods and traditional manual decision-making. As shown in Fig. 2, the hybrid GA rapidly improves profit in early generations and gradually converges, demonstrating stable optimization behavior across all stores.

While alternative metaheuristics such as Simulated Annealing (SA), Particle Swarm Optimization (PSO), and Bayesian Optimization (BO) are often utilized in price optimization, GA was selected for this framework due to its superior performance in high-dimensional discrete spaces. Unlike SA, which follows a point-to-point search trajectory that can be slow to navigate vast landscapes, the GA's population-based approach facilitates the simultaneous exploration of thousands of pricing strategies. Additionally, while BO is highly efficient for optimizing expensive functions with few parameters, it often becomes computationally prohibitive as the number of SKUs increases [12]. This parallel exploration is essential given the combinatorial explosion of the search space, which contains approximately unique combinations. The experimental results highlight the following key performance indicators:

- Superiority over Human Selection: The most significant finding is that the hybrid GA-XGBoost model achieved a profit uplift of up to 7% compared to historical Human Selection. This demonstrates that while human experts can provide reasonable price points, they often struggle to account for the complex, non-linear interdependencies between hundreds of SKUs simultaneously. The GA-driven approach successfully identified "hidden" profit opportunities by optimizing the entire portfolio.
- Superiority over Brute Force: Compared to the Brute Force approach, the hybrid GA exhibited vastly superior search efficiency. Within the same 4-hour time budget, the Brute Force method failed to converge on a profitable equilibrium, as it lacked the structured exploration and exploitation mechanisms inherent in the GA (see Figure 3).

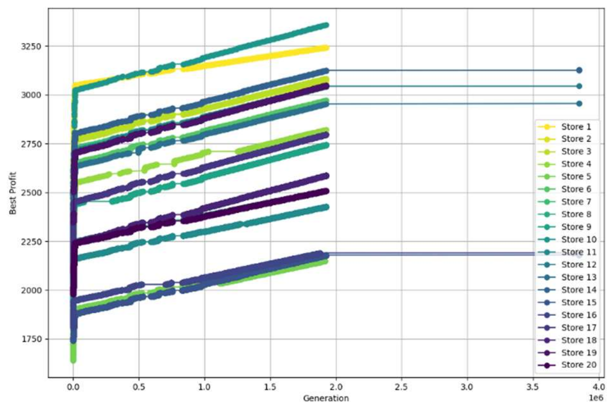

Fig. 3. Brute Force.

These results provide strong empirical evidence that a hybrid GA-XGBoost optimization framework can significantly enhance retail profitability, offering a 7% margin improvement over traditional manual strategies.

B. Combinatorial Explosion: Infeasibility of Brute Force Search

To further illustrate the necessity of the GA, we conducted a mathematical complexity analysis of the pricing search space. Given a portfolio of n = 100 SKUs, with each product having an average of M = 90 possible price points (within the ±30% constraint), the total search space is defined by the exponential function M^n.

The resulting number of unique pricing combinations is astronomical:

$$90^{100} \approx 2.65 \times 10^{195} \text{ combinations}$$

To put this into perspective, even if we utilized the world's most powerful exascale supercomputer capable of performing 10^{18} (one quintillion) price evaluations per second, the time required to exhaustively search for the global optimum would be approximately 8.4×10^{169} years.

This duration exceeds the current age of the universe (1.38×10^{10} years) by a significant magnitude. This combinatorial explosion confirms that a brute-force approach is fundamentally incapable of solving the multi-product price optimization problem in a retail environment.

In contrast, our hybrid GA-XGBoost model, through its intelligent exploration and exploitation mechanisms, converged on a superior pricing strategy within a mere 4-hour window. This comparison highlights the framework's efficiency in navigating high-dimensional spaces to find near-optimal solutions that would otherwise be computationally intractable.

VI. Conclusion and Future Work

This research presents a hybrid GA-XGBoost framework designed to optimize multi-product retail pricing by bridging the gap between predictive accuracy and operational profitability. By integrating an XGBoost-based demand predictor with a GA enhanced by cross-price elasticity, the system successfully navigates a high-dimensional search space on a scale that renders traditional brute-force methods impossible.

Experimental results demonstrate that the hybrid model significantly outperforms both stochastic search and manual pricing strategies. A key finding is that incorporating competitor pricing, inventory constraints, and cross-price elasticity as independent variables is crucial for modern retail optimization, achieving a profit uplift of up to 7% over human-driven decisions. Furthermore, the implementation of adaptive mutation to evade local optima and parallel processing for store-by-store decomposition ensures that the framework remains both robust and scalable.

Beyond the profit increase, this study reveals that cannibalization effects play an essential role in price optimization by accounting for these interdependencies. The model identifies pricing sets that maximize portfolio margins rather than individual product sales. While the system currently converges on near-optimal strategies within a practical 4-hour window, computational efficiency remains an important area for improvement. Ultimately, this work provides a tool for retail managers to maximize margins while respecting real-world market constraints.

Suggestions for future research are as follows.

- Early Stopping Strategy: To reduce resource consumption, a convergence-based cut-off can be implemented. The optimization will terminate if profit improvement falls below a 0.1% threshold in a certain time, preventing excessive runtime once the model reaches the point of diminishing returns
- Cross-Industry Application: Future research should apply this framework to other retail sectors, such as fashion or electronics. This will evaluate the model's robustness and generalizability beyond the grocery retail context.
- Matrix-Based Optimization: Future iterations will adopt a matrix-based batch processing approach to accelerate XGBoost demand forecasting and significantly reduce overall optimization time.

Acknowledgment

The authors would like to thank CP Axtra Public Company Limited for providing the proprietary retail datasets and technical insights that made this empirical study possible.

References

[1] S. Ito and R. Fujimaki, "Optimization beyond prediction: prescriptive price optimization," NEC Corp., arXiv:1605.05422v2, May 2016.

[2] M. Hemmati, S. M. T. Fatemi Ghomi, and M. S. Sajadieh, "Inventory of complementary products with stock dependent demand under vendor managed inventory with consignment policy," Scientia Iranica, vol. 25, no. 4, pp. 2347–2360, 2018.

[3] R. L. Das and R. K. Jana, "Some studies on EPQ model of substitutable products under imprecise environment," in Asset Analytics. Singapore: Springer, 2019, pp. 331–360.

[4] K. J. Talluri and G. J. Van Ryzin, The Theory and Practice of Revenue Management. New York: Springer, 2004.

[5] K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi, "Analytics for an online retailer: demand forecasting and price optimization," Manuf. Serv. Oper. Manag., vol. 18, no. 1, pp. 69–88, 2016.

[6] O. Besbes and A. Zeevi, "On the surprising effectiveness of predictive labeling in price optimization," Manage. Sci., vol. 61, no. 11, pp. 2792–2803, Nov. 2015.

[7] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, 2016, pp. 785–794.

[8] C. C. Peck and A. P. Dhawan, "Genetic algorithms as global random search methods: an alternative perspective," Evol. Comput., vol. 3, no. 1, pp. 39–80, 1995.

[9] M. Mitchell, An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.

[10] S. Uyar, S. Sariel, and G. Eryigit, "A gene-based adaptive mutation strategy for genetic algorithms," Lect. Notes Comput. Sci., vol. 3103, pp. 271–281, 2004.

[11] I. Rahman, P. Vasant, B. S. M. Singh, and M. Abdullah-Al-Wadud, "On the performance of accelerated particle swarm optimization for charging plug-in hybrid electric vehicles," Alexandria Eng. J., vol. 55, no. 1, pp. 419–426, Mar. 2016.

[12] H. F. Maathuis, R. De Breuker, and S. G. P. Castro, "High-dimensional Bayesian optimisation with large-scale constraints via latent space Gaussian processes," arXiv:2412.15679, Dec. 2024.