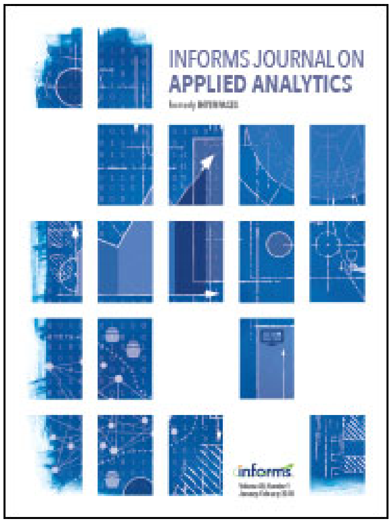


INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Alibaba Realizes Millions in Cost Savings Through Integrated Demand Forecasting, Inventory Management, Price Optimization, and Product Recommendations

Yuming Deng, Xinhui Zhang, Tong Wang, Lin Wang, Yidong Zhang, Xiaoqing Wang, Su Zhao, Yunwei Qi, Guangyao Yang, Xuezheng Peng

informs
https://pubsonline.informs.org/journal/ijaa

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 53, No. 1, January–February 2023, pp. 32–46
ISSN 2644-0865 (print), ISSN 2644-0873 (online)

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

Alibaba Realizes Millions in Cost Savings Through Integrated Demand Forecasting, Inventory Management, Price Optimization, and Product Recommendations

Yuming Deng,[a] Xinhui Zhang,[a,b,*] Tong Wang,[a] Lin Wang,[a] Yidong Zhang,[a] Xiaoqing Wang,[a] Su Zhao,[a] Yunwei Qi,[a] Guangyao Yang,[b] Xuezheng Peng[b]

[a] Alibaba Digital Supply Chain Divisio, Hangzhou, Zhejiang 311121, China; [b] Alibaba Freshippo Division, Hangzhou, Zhejiang 311121, China
*Corresponding author

Contact: yuming.dym@alibaba-inc.com (YD); maisui.zxh@alibaba-inc.com, https://orcid.org/0000-0001-9062-1190 (XZ); wt197481@alibaba-inc.com (TW); marvin.wl@alibaba-inc.com (LW); tanfu.zyd@alibaba-inc.com (YZ); robin.wxq@alibaba-inc.com (XW); sharpe.zs@alibaba-inc.com (SZ); yunwei.qyw@alibaba-inc.com (YW); guangyao.ygy@alibaba-inc.com (GY); shouqin.pxz@alibaba-inc.com (XP)

Accepted: October 5, 2022

https://doi.org/10.1287/inte.2022.1145

Abstract: Alibaba, which operates one of the world's largest e-commerce platforms, designed a comprehensive omnichannel retail infrastructure that enables both online and offline ordering of products, ranging from general merchandise to fresh produce. Intelligent decisions in the supply chain, such as demand forecasting, inventory management, and price optimization, are critical to the success of any retail business; however, many unique features in retail operations defy traditional operations research solutions and pose considerable challenges. Alibaba developed a series of models and algorithms, namely, deep-learning algorithms for demand forecasting, simulation optimization–based models for inventory management, price optimization for promotions, and markdown optimization for product recommendations. Alibaba has implemented these algorithms in almost all its retail businesses over the last three years and has generated, on an annual basis, $42 million of savings in shrinkage and inventory costs, $110 million in increased sales, and $13 million dollars in increased profit.

Keywords: demand forecasting • inventory management • price optimization • product recommendation • simulation optimization • Edelman award

Introduction

Alibaba, which operates one of the world's largest e-commerce platforms, has actively driven technology innovation through the digital transformation of businesses, from commerce to logistics, and has designed a comprehensive omnichannel retail infrastructure to enable both online and offline ordering of products, ranging from general merchandise to fresh produce. Retail is the cornerstone of any economy, and this retail infrastructure will be the foundation and pillar of Alibaba's businesses in the future.

Based on service radius and diversified fulfillment modes, the retail infrastructure includes various Alibaba business subsidiaries, including Freshippo, Sun Art, TMall Mart, TMall Global, and TMall (Table 1). In this table, we categorize the service areas by service radius and delivery modes, such as neighborhood areas of 3 km in radius, served via 30-minute or one-hour delivery, citywide areas of 20 km in radius via same-day or next-day hybrid fulfillment modes, and nationwide areas served via one-day to three-day hub-and-spoke delivery systems. Hu et al. (2022) provide details of these business subsidiaries.

Problems and Challenges

Alibaba has been at the forefront of innovating supply chain and operations technology as it builds its infrastructure and develops retail models. In the 2022 Edelman competition video (Alibaba 2022), Daniel Zhang, chairman and chief executive officer (CEO) of Alibaba Group, stated "no matter how retail business models change, the fact that their core competencies reside in supply chain technology does not change" and "scientific and intelligent decisions are critical to the success of the retail business."

However, these decisions can be more complex in an omnichannel retail environment with the addition of online websites and mobile applications. In essence, these decisions can be categorized into three areas: (1) supply chain management, such as demand forecasting, inventory management, pricing, and markdowns; (2)

Table 1. Alibaba's New Retail Infrastructure Is Built on Different Service Radii and Diversified Fulfillment Modes and Will Be the Foundation and Pillar of Alibaba's Businesses in the Future

Fulfillment modes	Neighborhood (3 km)	Citywide (20 km)	Nationwide
Thirty-minute or one-hour instant delivery	Freshippo		
Same-day/next-day delivery hybrid fulfillment		Sun Art TMall Mart	
One- to-three-day delivery hub-and-spoke system			TMall TMall Global

recommendation systems, such as the selection and ranking of products to match against customers' preferences; and (3) logistics management, such as network design, warehouse operations, and delivery operations. We describe our vehicle routing algorithms, which solve a variety of warehouse picking and last-mile delivery problems in logistics management, in Hu et al. (2022). In this paper, we focus on the models and algorithms we developed for the first two areas (i.e., supply chain management and recommendation systems) to better match supply and demand. Recommendation systems provide an additional means to match products to customers, serve a role similar to the one brick-and-mortar stores serve for traditional retail, and are just as critical in the omnichannel retail environment.

There has been an increase in the study of omnichannel retail in recent years (see Harsha et al. (2019), Feldman et al. (2022), and Wan (2022) for studies of various aspects of this retail environment). Nevertheless, in the process of deriving technical solutions, we still encounter many problems and challenges, both old and new, that the operations research (OR) community has not yet solved or implemented broadly. We present specific challenges here.

(1) *Demand affected by various factors and interactions among products.* The demands of many products exhibit seasonal patterns across geographical regions and are affected by promotions, marketing campaigns, and interactions among products. For example, changes in the price of pork may affect beef sales because consumers view beef and pork as alternative sources of protein (i.e., they are substitute goods), whereas changes in the price of tortilla chips may affect salsa sales because consumers often buy them together (i.e., they are complementary goods). Forecasting in retail requires automated, practical, and yet, interpretable procedures to learn these interactions among a substantial number of related products, and thus has posed considerable challenges to traditional forecasting approaches.

(2) *Complex inventory operations*. Although most inventory models are based on some distributional assumption for demand, such as Normal or Poisson, this assumption may not hold in practice, for example, due to distinctively different online and offline demand streams. Furthermore, inventory policies may need to be customized based on the type of products and operational practices. In the case of perishable products (e.g., fresh produce that have short life cycles), they are subject to age-dependent sales and inventory issuing patterns. For example, in-store purchases by customers might exhibit a last-in-first-out (LIFO) pattern, while online orders by pickers might follow a first-in-first-out (FIFO) pattern. Dramatic differences exist for deriving optimal inventory policies under these different patterns, and thus require novel approaches beyond traditional inventory models.

(3) *Markdowns to be coordinated in the online and offline channels*. In practice, perfectly matching inventory and demand is impossible, even when using state-of-the-art forecasting algorithms and an optimized inventory policy; as a result, price markdowns are often used in stores to reduce the excess inventory at the end of its season or life span. For brick-and-mortar stores, these markdowns are available to everyone who sees them. However, in omnichannel retail, coordinating online and offline markdowns (i.e., what products to mark down, what prices to set for the marked-down products, when to begin the markdown process, and the ranks of markdown products in the list of recommendations for each customer) provide levers or opportunities to mitigate these mismatches. These large-scale online optimization problems are extremely challenging and must be solved in real time. The solution of these problems nevertheless could provide significant and surprising financial benefits, which we should never underestimate.

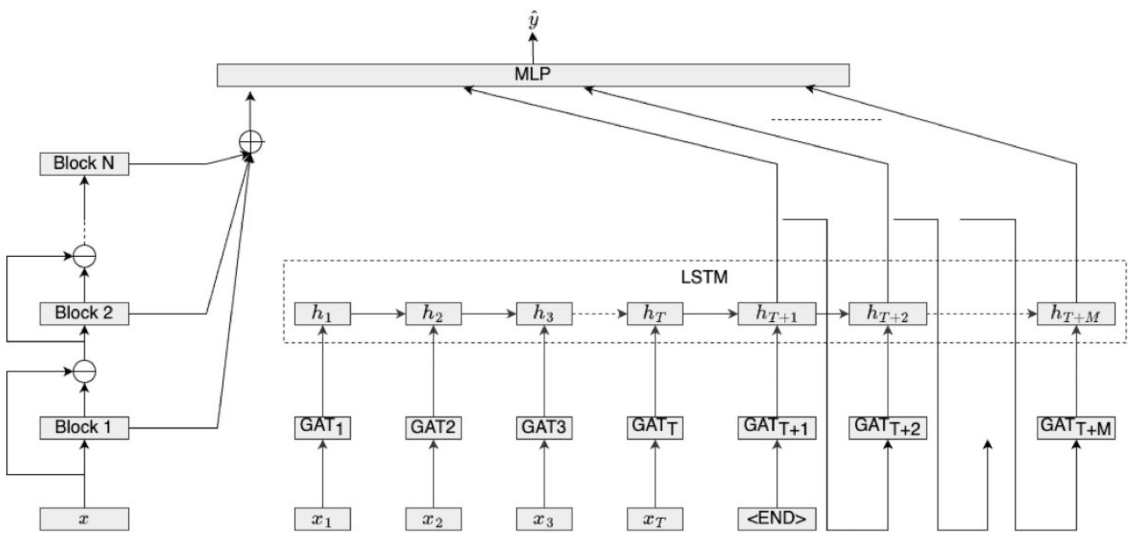
Technical Solutions

To address these challenges, the OR teams at Alibaba developed a series of models and algorithms, including deep learning–based algorithms for demand forecasting, and simulation optimization–based models for inventory management, price optimization, and real-time markdown pricing and product recommendations.

Demand Forecasting

Falcon, our deep learning–based forecasting algorithm (Figure 1), is built on the concepts of the most recent developments reported in the literature, such as DeepAR (Salinas et al. 2020), Prophet (Taylor and Letham 2018), Wavenet (van de Oord et al. 2016), and N-BEATS (Oreshkin et al. 2019), to generate time-series forecasts for demands. Falcon includes two main branches; the left branch explores the time series components such as

Figure 1. Our Deep Learning–Based Forecasting Algorithm, Falcon, Consists of Two Main Branches



Notes. The left branch is a stack of additive blocks, and the right branch is a spatial-temporal graph attention (STGAT) network. We use the following notations in the figure: MLP, multilayer perceptron; LSTM, long short-term memory; h_t, the hidden state for time t inside LSTM; GAT, graph attention network; x_t, the input feature vector of time t.

trend and seasonality, and the right branch explores the interactions among products.

The left branch is an additive structure with multiple blocks stacked upon one another; each block, modeled as a subnetwork, captures a specific component (e.g., trend, seasonality) of the time series. These blocks are connected through a residual network and merged to obtain the forecast. The right branch uses graph attention networks (GATs) to discover product interactions and a long short-term memory (LSTM) network to connect the GAT blocks and thus capture the changes over time. It is, in essence, a spatial-temporal network to generate time series forecasts. The outputs from the left and right branches are assembled to create the final forecast through a multilayer perceptron (MLP).

This architecture overcomes the shortcomings of traditional statistical methods (e.g., the inability to capture product features and interactions among products), and of machine learning–based methods (e.g., the inability to explicitly exploit the time series structures). This approach has demonstrated a significant improvement in forecast accuracy compared with state-of-the-art forecasting algorithms and has exhibited superiority in interpretability and simplicity, as we explain later.

Left Branch: Decomposition Through the Selection of Neural Network Blocks

Falcon provides a rich set of blocks to model the effects of various factors such as trend, seasonality, pricing, and events such as holidays and promotional campaigns, which are essential in a retail context. Figure 2 illustrates a selected set of representative neural network blocks, such as trend, seasonality, and event blocks. The blocks share a similar structure in the lower half; the input data

Figure 2. Falcon Provides a Rich Set of Blocks That Capture the Trend, Seasonality, and Event Effects to Improve Forecast Accuracy and Interpretability (Left Branch)

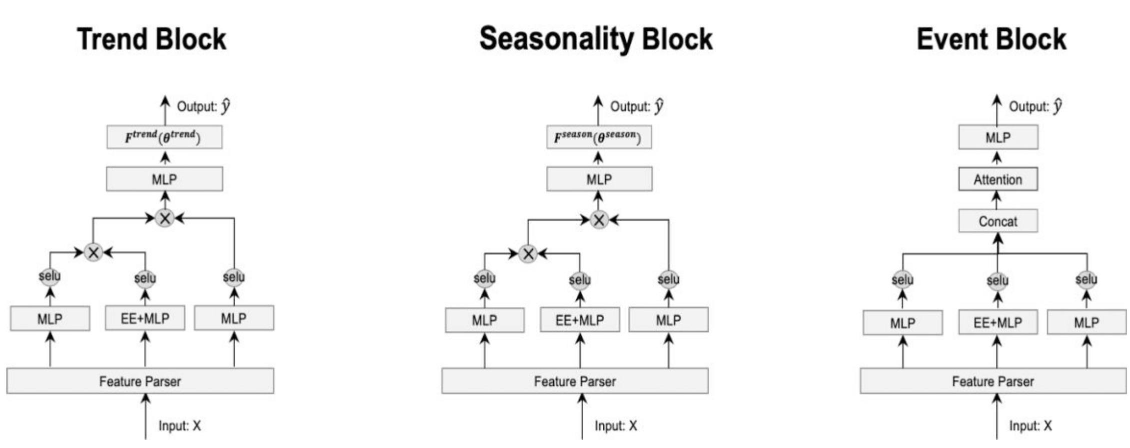


Figure 3. Forecast with (Linear) Trend Block Is a Straight Line but Does Not Show Seasonality and Market Event Patterns

are parsed and encoded in various ways. The difference lies in the upper half. The trend and seasonality components of the time series can be effectively extracted by fitting specific shape functions such as polynomial ($F^{trend}()$ in Figure 2) and Fourier series ($F^{season}()$ in Figure 2), similar to those in N-BEATS (Oreshkin et al. 2019) and Prophet (Taylor and Letham 2018). In the event block, a temporal attention block is designed to extract the impact of a marketing campaign on the time series.

The decomposition of time series into components, modeled as stacks of additive blocks, allows us to examine each component's effect separately and enhance the forecasts' interpretability. Figures 3–5 demonstrate the impact of adding blocks sequentially. In Figure 3, only one linear trend block is used in the neural network, and we observe the forecast as a downward-sloping straight line. In Figure 4, with the addition of a seasonality block, the forecast now captures the annual seasonal cycle with an initial increasing and then decreasing pattern. Finally, in Figure 5, where an event block is added to capture the impact of marketing events, the forecast now exhibits intermittent ups and downs based on the timing of these events. Using these figures as illustrations, we can explain in a straightforward way the contribution of each block/component and how the forecast is eventually generated.

Right Branch: Spatial-Temporal Graph Attention Network for Product Interactions

The right branch of the network (Figure 1) captures the interactions among products, and as we illustrate in Figure 6, its architecture is a spatial-temporal graph attention network (STGAT), where related products are modeled as nodes. The spatial connections in the network represent the interactions among products, whereas the temporal connections represent the time series component. Various methods, such as shopping cart analysis, can be used to identify related products to reduce the number of edges between nodes in the graph and thus to improve computational efficiency.

Improvement in Forecasting Accuracy

Table 2 summarizes the result of an online experiment that we conducted to benchmark Falcon against other popular

Figure 4. Forecast with (Linear) Trend and Seasonality Blocks Exhibits a Downward Seasonal (i.e., an Up and Down) Pattern

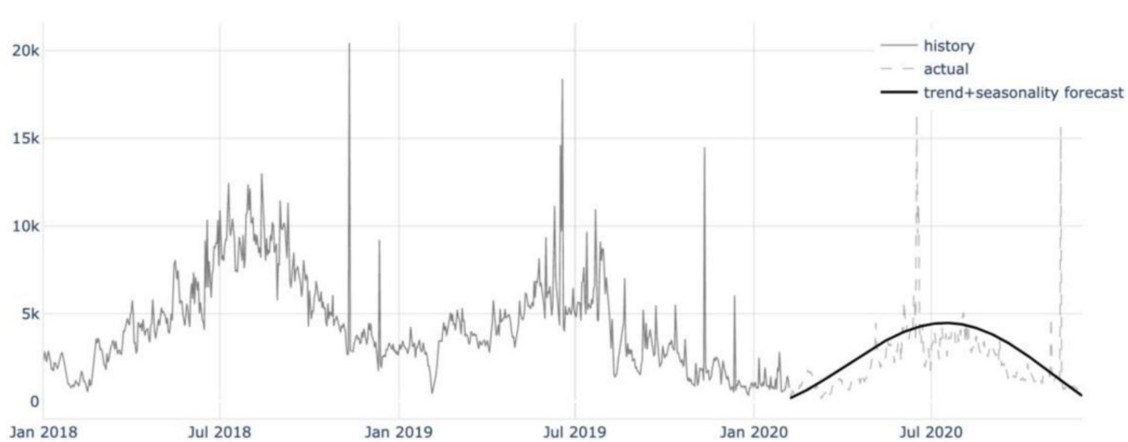
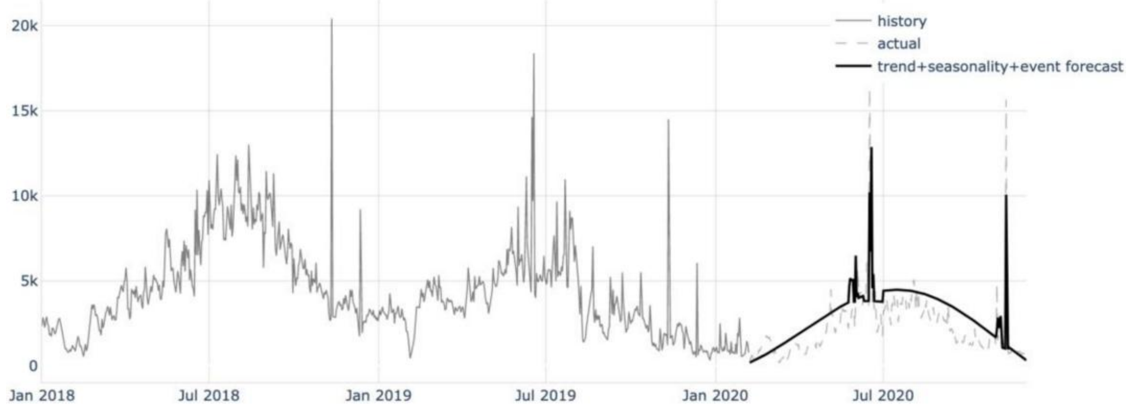


Figure 5. Forecast with (Linear) Trend, Seasonality, and Event Blocks Captures the Trend, Seasonality, and Intermittent Jumps Resulting from Marketing Events



algorithms, including DeepAR, Prophet, Wavenet, and the statistical method MA28 (moving average of 28 days), for a set of more than 20,000 products sold at TMall Mart over a period of three months from September to December 2021. The results show that Falcon exhibits superior forecasting accuracy against the benchmarks. Here, accuracy is defined as "1−WMAPE," where WMAPE represents weighted mean absolute percentage error, and the weights are the actual sales of the products. Interestingly, Prophet performs worse than a simple moving average. We believe the reason is that Prophet tends to overfit the data and treat sudden fluctuations as systematic trends. In our experiments, we identified multiple dramatic fluctuations, which were due to major marketing events but not necessarily systematic trends, and thus caused Prophet's poor performance.

The deployment of the algorithm resulted in improved forecasting accuracy, ranging from 2% to 10%, in various Alibaba subsidiaries, which include Freshippo, Sun Art, TMall Mart, TMall Global, and TMall. In addition, we found that the algorithm is particularly effective in forecasting new products, which is always challenging because of the lack of historical information. In one subsidiary, Freshippo, although the overall forecast accuracy improved by only 1.65% across all products, the forecast accuracy for new products increased by 7.96%. This improved accuracy occurred because, although sales history for the new products does not exist, the algorithm can reconstruct and extrapolate backward in time creating this history based on the related existing products.

Inventory Optimization

Inventory optimization is a well-studied area in OR; however, in practice, its implementation could be limited because of practical challenges, such as empirical demand distributions and complex inventory consumption patterns,

Figure 6. Right Branch is a Spatial-Temporal Graph Attention Network Structure Interconnected by a Long Short-Term Memory Network

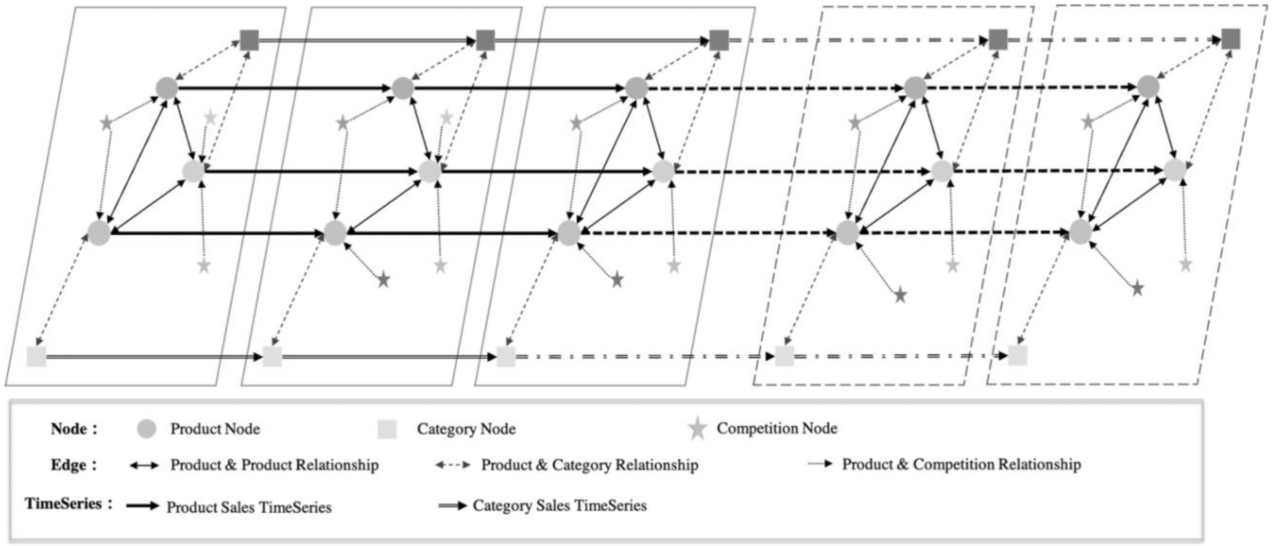


Table 2. Comparison of Various Forecasting Methods Shows That Falcon Exhibits Superior Accuracy

	Falcon	DeepAR	MA28	Prophet	Wavenet
WMAPE	52.22%	37.39%	35.77%	29.41%	43.38%

in the form of either age-dependent sales or inventory issuing patterns, as we discuss here.

Demand Distributions. In many theoretical inventory models, demand is assumed to follow a known distribution; however, this may not be the case in practice. For example, in an omnichannel retail environment such as Freshippo, there are two sources of demand, online and offline channels, which follow different patterns (Figure 7, left). The combination of the demand from these two channels (Figure 7, right) may not follow a unimodal distribution. In this figure, the vertical axis represents the empirical probability that a specific amount of demand will be realized, and the horizontal axis represents this amount of demand.

Inventory Consumption Patterns. A perishable product's inventory level, especially of fresh produce, as mentioned, is affected heavily by the age-dependent sales patterns by the customers and inventory issuing patterns or the overall inventory-consumption patterns. For example, customers at brick-and-mortar stores may choose the freshest products, a LIFO pattern, whereas pickers for online orders may choose the earliest arrivals on the shelf, a FIFO pattern, or randomly pick a product on the shelf because of time constraints to pick. Our empirical observations and computational simulations indicate dramatic differences in terms of inventory levels, shrinkage levels, and out-of-stock levels (unfulfilled demand) among the various inventory consumption patterns.

As an example, Figure 8 shows the sales for ginger (in packs) in a store over one month. The vertical axis represents the sales, and the horizontal axis represents the calendar day. Assuming a policy of periodic review base stock policy (S, T) where S is the order-up-to stock level and T is the length of the review period, the optimal solution from simulation-optimization for a FIFO pattern is (24, 1) with an average on-hand inventory level of 6.7 packs, shrinkage level of three packs, and out-of-stock level of three packs. The optimal solution from simulation-optimization for a LIFO pattern is (16, 1), with an average on-hand inventory level of 2.4 packs, shrinkage level of eight packs, and out-of-stock level of 10 packs. We show the ending inventory under these two scenarios in Figures 9 and 10.

Although it is not surprising that the inventory levels under LIFO (i.e., picking the freshest product first) could be more costly in terms of shrinkage and out-of-stock costs than those of FIFO, the difference is staggering. Thus, inventory consumption patterns are particularly critical to effectively managing inventory for products with short life cycles; yet such patterns are hard to model using an analytical approach. Moreover, in practice, a store's actual inventory consumption patterns can be a mixture of LIFO and FIFO.

Solution Approach. A simulation-based optimization (SOPT) approach is adopted to model and solve complex inventory problems under different network structures,

Figure 7. Combined Demand for Online Channels (Top Left) and Offline Channels (Bottom Left) Form a Multimodal Demand Distribution for an Omnichannel Environment (Right)

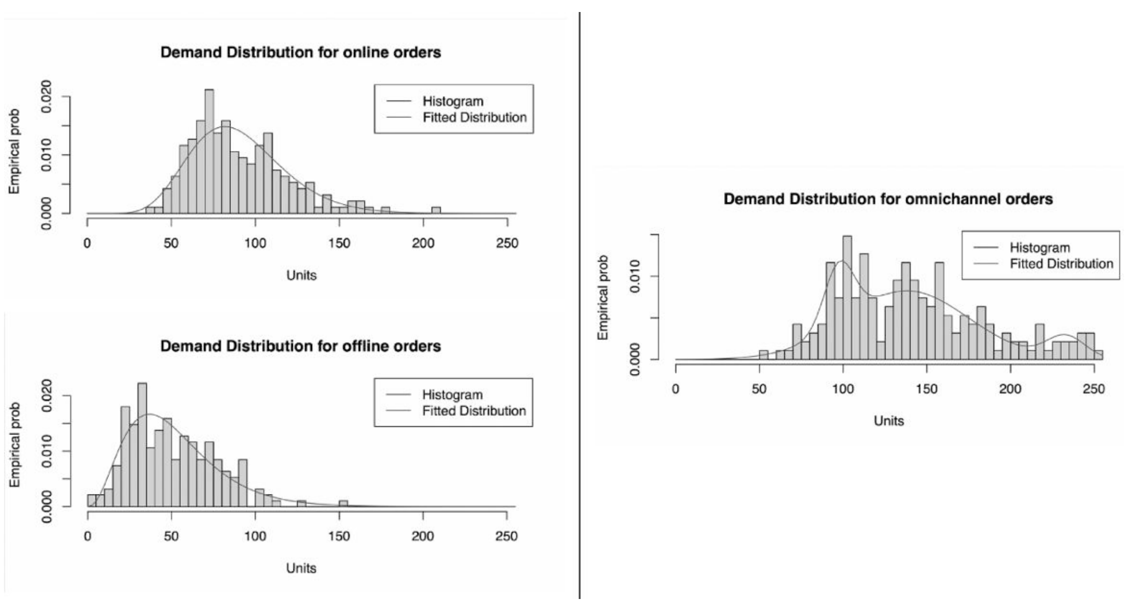
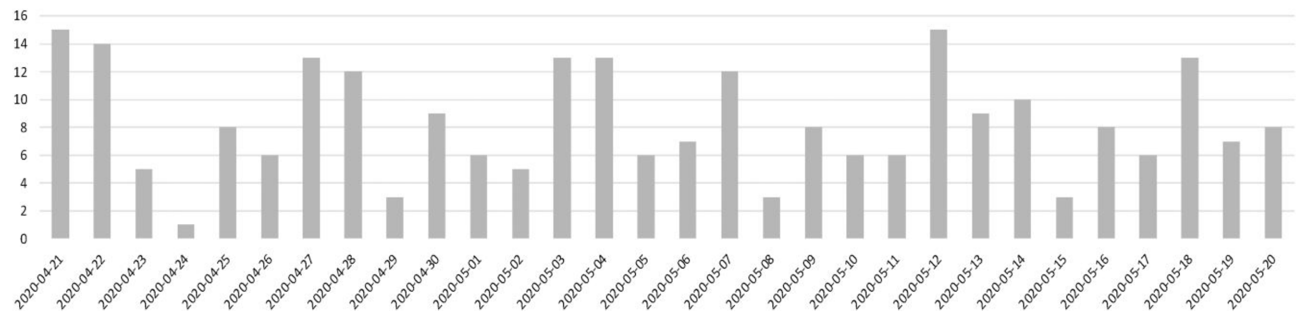


Figure 8. Sales for Ginger over a Month Have a Mean of 8.3 Packs and a Shelf Life of Three Days



product life cycles, inventory policies, and inventory consumption patterns. We believe that the SOPT approach offers significant advantages to solving complex inventory problems, as other work in the retail domain (Fu and Healy 1997, Zhang et al. 2014) also indicate.

In general, an SOPT algorithm uses simulation to model complex inventory operations in the real world and applies optimization to derive the best inventory policies. However, the SOPT process could be slow due to a large number of solution evaluations via simulation. In our implementation, we used extensive output data from the simulation (e.g., minimum inventory levels and maximum inventory shortages over the simulation period) to guide the search for a solution in the optimization process, as illustrated later in the neighborhood moves. Using this information has significantly improved the computational efficiency of the SOPT algorithm.

In summary, our SOPT algorithm follows an iterative approach. First, the system takes configuration parameters such as demand profiles, product shelf lives, inventory consumption patterns, calendar-based ordering patterns (defined in the next paragraph), and inventory policies as input to generate scenarios for simulation. The algorithm then iterates through the simulation and optimization procedures as follows. The simulation procedure is used to evaluate the current solution on a given set of scenarios over the simulation period and to generate the objective value and other information related to the current solution, such as minimum inventory levels and maximum inventory shortages over the simulation period. The optimization selection procedure chooses, either randomly or adaptively, an appropriate optimization method for improvement from a pool of candidates such as local search and response surface, to be used in the optimization procedure to generate the next solution. The system iterates until the stopping criterion is met after which it generates the best solution found.

Simulation Procedure. Simulation offers significant flexibility and could be adapted easily to mimic complex inventory problems in realistic settings. Our simulation allows for the configuration of various components in an inventory system. Examples include (1) inventory policies such as periodic, (s, S) or (S, T), or continuous, (Q, R), where s represents the reorder point, S represents the order-up-to level, T represents the length of the review period, Q represents order quantity, and R represents the reorder point; (2) inventory consumption patterns such as FIFO, LIFO, LIFOR(r), FIFOR(r), or any empirical inventory consumption patterns, where LIFOR(r) and FIFOR(r) are variations of LIFO and FIFO, and r represents the minimal residual shelf life required (Haijema et al. 2007); (3) calendar-based ordering patterns, such as ordering on fixed days of the week (e.g., Mondays, Wednesdays, and Fridays, or Tuesdays and Thursdays);

Figure 9. Ending Inventory over a Month Under FIFO with an (S, T) Policy Is (24, 1) and Has an Average On-Hand Inventory of 6.7, Shrinkage of 3, and Out-Of-Stock Level of 3

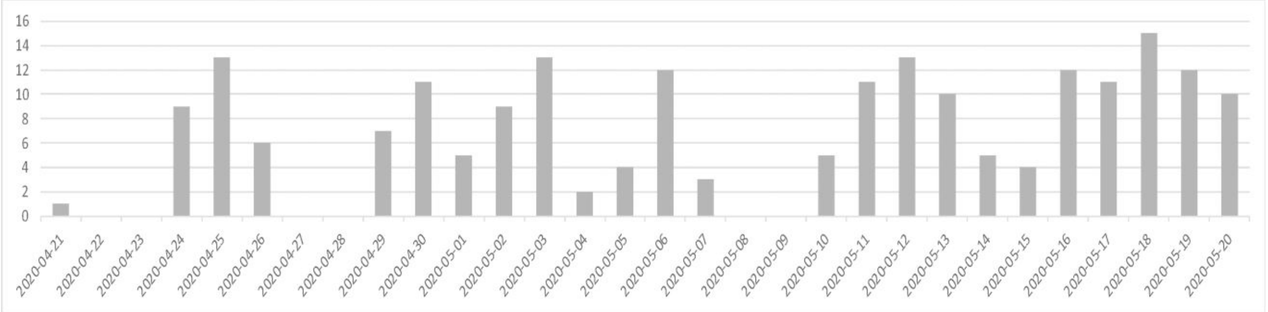
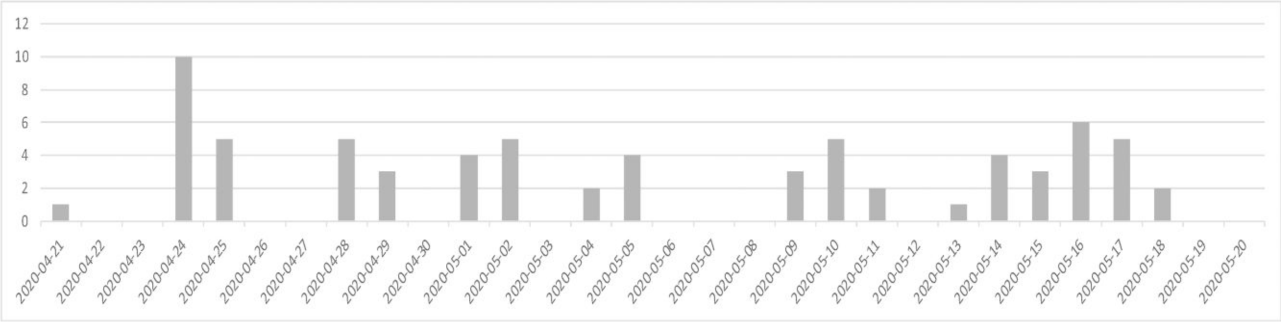


Figure 10. Ending Inventory over a Month Under LIFO with an (S, T) Policy Is (16, 1) and Has an Average On-Hand Inventory of 2.4, Shrinkage of 8, and Out-Of-Stock Level of 10

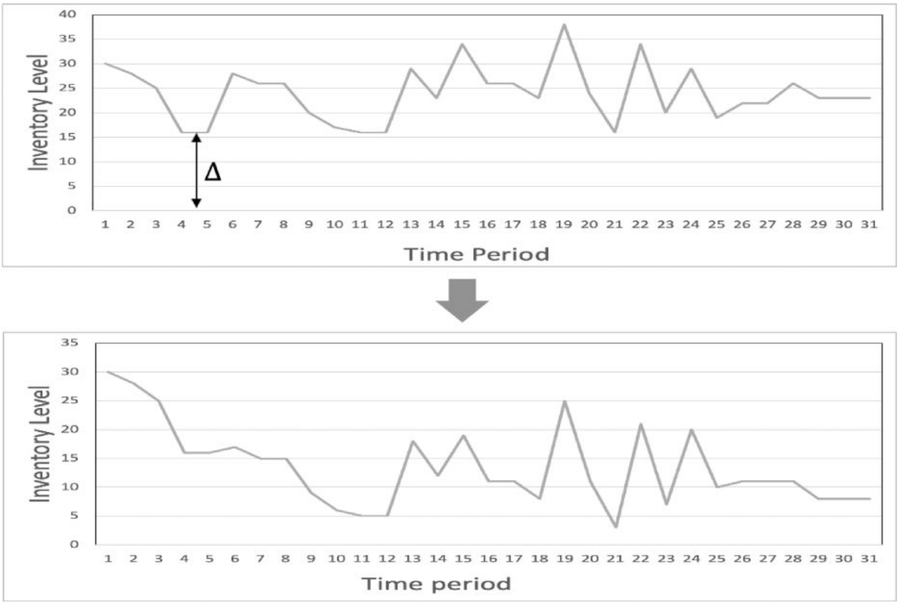


(4) supplier performance in delivery lead time and fill rate; and (5) demand patterns that include parametric distributions, such as Normal or Gamma, and nonparametric empirical distributions. Upon the termination of the simulation, the solutions, their objective function values, and various output data, as discussed previously, such as minimum inventory levels and maximum inventory shortages, from the simulation are returned to be used later in the optimization procedure to speed up computational times.

Optimization Procedure. The optimization procedure integrates multiple optimization methodologies such as metaheuristics and response surface methodologies. We put significant effort into developing techniques to speed up the computational times of our SOPT algorithms. These techniques include (1) the adaptive selection of metaheuristic algorithms, such as random descent, metropolis search, tabu search, and variable neighborhood search, and (2) the development of inventory-specific neighborhood move operators that use the characteristics of the solutions to guide the search for solutions in promising regions.

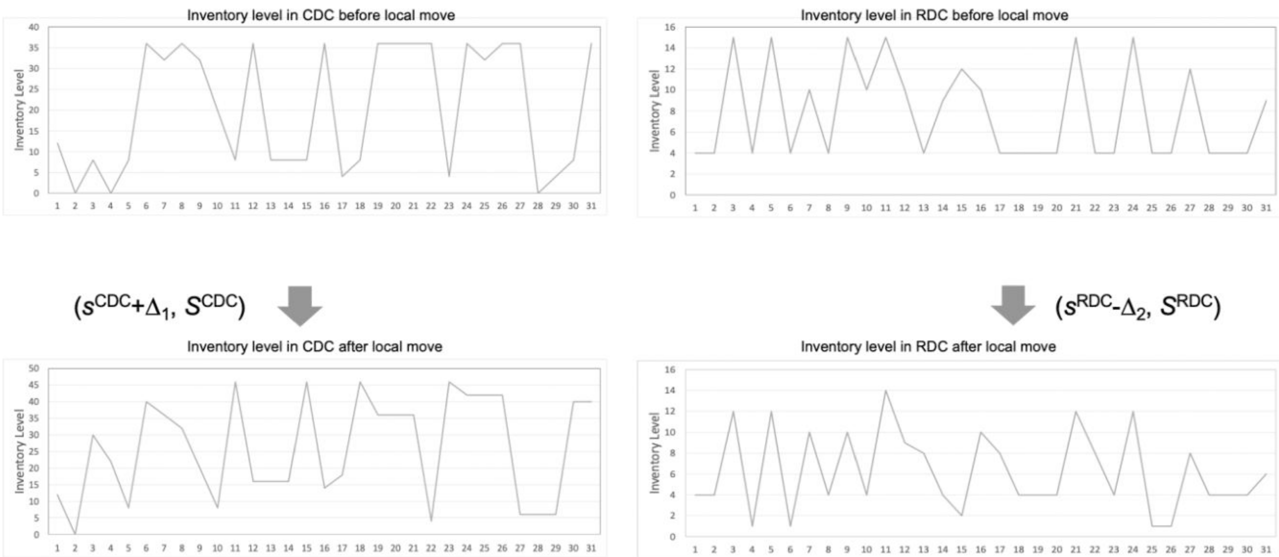
We show an example of a local neighborhood move on the simulation output (ending inventory) of an (s, S) inventory policy for a single echelon in Figure 11. In the top subfigure, the simulation shows a positive minimum ending inventory (Δ) across the simulation horizon, suggesting that further reducing s by Δ could lead to lower inventory levels without affecting the out-of-stock instances. A local candidate move, by reducing s by Δ, leads to the solution of (s–Δ, S), and the simulation for (s–Δ, S) is shown in the bottom subfigure of Figure 11, where the ending inventory is reduced without any

Figure 11. Inventory-Based Local Neighborhood Move for the Case of a Single-Echelon-Single-Item Problem



Notes. The top row shows inventory levels of an item under an (s, S) policy and the bottom row shows inventory levels of the item under an (s–Δ, S) policy. The positive minimum ending inventory (Δ) in the top row suggests reducing s by Δ, which leads to a neighborhood move to (s–Δ, S). Nevertheless, due to stochastic lead times, the ending inventory under (s–Δ, S) may not always reach zero.

Figure 12. Inventory-Based Local Neighborhood Moves for a Two-Echelon-Single-Item Problem



Notes. The top row shows inventory levels at the CDC (upper left) and RDC (upper right) before a pair of local moves to move excess inventory from the RDC to the CDC. The bottom row shows inventory levels at the CDC (lower left) and RDC (lower right) after the local move. s^{CDC} represents the order point at the CDC, S^{CDC} represents the order-up-to level at the CDC, Δ_1 represents the step size of a local neighborhood move for the figures on the left, and Δ_2 represents the step size of a local neighborhood move for figures on the right.

out-of-stock instances. Due to the stochastic nature of simulation (e.g., different order calendars leading to different delivery dates) the ending inventory may or may not fall to exactly zero.

We next demonstrate an example move on the simulation output of a single item under a two-echelon network in Figure 12. The central distribution center (CDC) is the first echelon, and the regional distribution center (RDC) is the second echelon. The top subfigure of Figure 12 shows the ending inventories in the CDC (left) and the RDC (right) of the solution. As we can see, these results show a positive ending inventory at the RDC, yet a tight inventory level at the CDC, suggesting that it is possible to move inventory from the RDC to the CDC through a pair of local moves, from (s^{CDC}, S^{CDC}), (s^{RDC}, S^{RDC}) to ($s^{\mathrm{CDC}}+\Delta_1$, S^{CDC}), ($s^{\mathrm{RDC}}-\Delta_2$, S^{RDC}). Moving inventory from the RDC to the CDC can also enable better risk pooling. We show the simulation results of the new solutions after the local moves in the bottom subfigure of Figure 12.

Computational Results. To reduce computational time and increase computational efficiency, we adopted many techniques, such as sampling techniques, to initially generate only a subset of scenarios and then dynamically include additional scenarios; nevertheless, the inventory-specific move approach proved to be the most effective. We conducted various experiments to demonstrate the effectiveness of the SOPT approach, two of which we present here.

The first experiment was to evaluate the performance of the SOPT approach on parametric distributions with known parameters, such as Poisson and Geometric distributions, against some well-known policies discussed in the literature, such as the myopic two-period policy (Gijsbrechts et al. 2022). For these distributions with a demand arrival rate of four, a holding cost of \$1 per unit per unit time, and a lost-sale cost of \$4 per unit, the SOPT approach yielded solutions with total costs that were on average 0.12% and 1.92% lower for the Poisson and Geometric distributions, respectively, compared with the myopic two-period policy. This experiment is simply to demonstrate that the SOPT approach can achieve a similar or near-optimal solution for demand patterns with a known distribution and known parameters. For nonparametric demand patterns with an empirical distribution, our algorithm shows better performance; see the second experiment.

The second experiment was to evaluate the SOPT approach's performance on empirical distributions from TMall Mart against a classical solution (Chen et al. 2022), an (s, S) policy, assuming Normal demand distribution. In the case of single-echelon inventory problems, the SOPT approach resulted in an increase of 1.3% in service levels and a decrease of 37.4% in inventory. In the case of two-echelon inventory problems, the SOPT approach reduced the overall inventory levels by 33% while maintaining the same service levels. In addition, the SOPT approach provides the ability to model complex inventory problems, as in the case of perishable products with short life cycles, which otherwise would be computationally intractable. We discuss these results under the *Implementation* section in the *History, Implementation, and Financial Benefits* section.

Price Optimization

Price optimization is critical to retail revenue management in two principal areas: (1) setting promotional prices in marketing campaigns to increase sales, which we do every two weeks at TMall Mart, and (2) setting markdown prices for inventory clearance to reduce shrinkage, which we do multiple times a day at Freshippo. The problem in setting promotional prices is to determine the promotional price for a set or category of products to maximize the total revenue while maintaining a target profit margin, although maximizing the total profit while maintaining a target revenue is also possible. We model this problem as a multiproduct promotional pricing problem, which we describe below in this section. The problem in setting markdown prices is to determine the prices for perishable products to reduce shrinkage or increase profit based on the products' real-time inventory. We developed wireless display tags for Freshippo such that prices could be pushed to these tags in real-time within stores. In the *Integrating Markdown Price Optimization into Product Recommendations* section, we discuss the innovative, integrated markdown price optimization and product recommendation system we developed for Freshippo.

Developing the Price and Demand Relationship. The relationship between price and demand plays an essential role in these decisions; yet, many challenges exist to accurately modeling this relationship. The most significant is the existence of confounding factors (e.g., promotional campaigns, holidays, weather). For example, a marketing campaign, which attracts more visitors to a website than on normal days, stimulates sales of all products including even those whose prices do not change. In this case, the marketing campaign acts as a confounding factor of sales.

To develop the relationship between price and demand to address these challenges, we apply causal inference (CI), which aims to infer the quantity change induced by a treatment such as a price change while considering the other variables as confounding factors. We use double machine learning (DML), a CI method, which assumes that there is a machine learning model to forecast the daily sales quantity, $q = g(X)$, and another model to forecast daily price, $p = f(X)$, where X is the vector of factors that are identical and simultaneously affect price and demand. CI estimates price elasticity, $\delta q/\delta p$, using the causal model $\delta q/\delta p = (q - g(X))/(p - f(X)) = h(X)$. With the elasticity prediction $h(X)$, the predicted sales are then given by $h(X) \times (p - f(X)) + g(X)$ for a given price p. Our computational results showed that DML models outperformed other forecasting models that do not consider the impact of price changes and produced interpretable price-demand relationships that are intuitive and in line with our expectations. Chernozhukov et al. (2018) provide details on DML.

Multiproduct Promotional Pricing Problem. Multiproduct promotional campaigns include various types of promotions; examples include 20% off, buy one get one free, and 30% off for orders of $100 or more. The challenge is to determine the set of promotions (i.e., campaign activities) for each product that will maximize revenue while maintaining a target profit margin. The corresponding mathematical model is shown in Appendix A, and we developed a metaheuristic, specifically, an evolutionary algorithm (Crepinsek et al. 2013), to solve it effectively.

Results. In our computational experiments, we first aimed to compare the accuracy for price and demand relationship using DML, through forecast accuracy, measured by the "1-WMAPE" metric. DML outperformed the simple price elasticity models, such as linear price and log-log elasticity models, and increased forecast accuracy of predicted sales by more than 6%. We then ran numerical experiments to investigate the performance of the metaheuristic for solving the multiproduct price optimization problem. For a marketing campaign with 10,000 products, each with 10 candidate prices, the metaheuristic could effectively solve the optimization problem in less than 100 seconds within 95% of optimality. Prior to this solution, human decision makers at TMall Mart made all promotional campaign decisions. Our online computational tests showed a 7% increase in sales, while maintaining the same profit margin, compared with the results generated by the human decision makers.

Integrating Markdown Price Optimization into Product Recommendations

In traditional brick-and-mortar stores, markdowns are typically available to everyone who sees them; however, in omnichannel retail, it is possible to integrate markdown optimization into product recommendations to steer customers toward products that have excess remaining inventory to increase the products' exposure. Although recommendation systems are common in online applications, integrating markdown optimization and product recommendations is a challenging and important problem in omnichannel retail operations and has only recently started to receive attention in the literature. Feldman et al. (2022) provide related studies.

Hierarchical Approach. To address our main problem of integrating markdown optimization and product recommendations, we aim (a) to set the markdown prices for products for online and offline channels at specific times of a planning horizon and (b) to determine the rank of products in the list recommended for a customer. In essence, if a product is ranked further down the ranking list, it is essentially equivalent to not displaying the product to a customer; if a product is ranked high on that list, it is equivalent to steering customers toward the product. Because of the problem's large size and real-time nature, we developed a three-step hierarchical approach, which we describe here, to solve this large-scale optimization problem.

Step 1: Over a given planning horizon, determine markdown prices for both online and offline channels and the associated sales based on price-demand relationships to minimize shrinkage or maximize revenue.

Step 2: For a short, typically three- to five-minute time interval, sample potential customers; for example, sample customers from the past five-minute interval from the same five-minute interval yesterday and from the same period on the same day one week ago. Then determine whether (or not) to display or rank a product at its suggested price to meet the expected sales derived from Step 1. The solution to the relaxation of this integer program to determine the rank provides a dual price to be used in Step 3.

Step 3: In real time, use the dual price from Step 2 to determine a ranking score for the product-ranking model.

The optimization problem in Step 1 can be modeled as a multiperiod pricing problem (see Perakis and Sood (2006) for an example). The output of this optimization provides the products to be optimized and their markdown prices and expected sales, which are sent as input to the optimization problem in Step 2. The optimization problem in Step 2 is the most challenging problem in the three steps because it must typically be solved frequently (every five minutes in our case); thus, it must be solved hundreds of times per day per store. The problem is to select a personalized set of products to display to each customer to maximize a specific metric, such as revenue, while satisfying various resource limitation constraints. The dual price for a product is used in online recommendations for each customer. We present the detailed model and the formulation in Appendix B; however, because of space limitations, we do not delve into the details of the algorithm for Step 3 (see Agrawal et al. (2014) for similar studies).

Results at Freshippo. At Freshippo, this integrated model determines the markdown discounts of perishable products for both online and offline channels on an hourly basis and dynamically adjusts the rankings of the products based on the available inventory as the customer browses the mobile applications. During our experiments, the algorithm was called more than 60,000 times per day, once every five minutes over 20 operating hours across 300 stores, and significantly improved revenue and profit, providing a 3% reduction in the scrap rate (i.e., the ratio of the salvage inventory value plus the amount of discount loss to the revenue). It is our belief that integrating markdown price optimization into product ranking recommendations provides a unique opportunity and a novel and effective way to increase revenue and reduce costs in an omnichannel retail environment.

History, Implementation, and Financial Benefits

History of OR at Alibaba

Figure 13 illustrates a brief history of OR at Alibaba.

Phase I: In 2014 Yuming Deng, the first research scientist in Alibaba's Supply Chain division, introduced OR to Alibaba. We established a supply chain OR team and started researching models for demand forecasting, inventory management, price optimization, and product recommendations. Although the initial models were relatively traditional, they effectively solved many supply chain problems.

Phase II: In 2017, OR was widely applied in supply chain applications, and we started researching deep-learning methods for forecasting and simulation-based optimization approaches for inventory management.

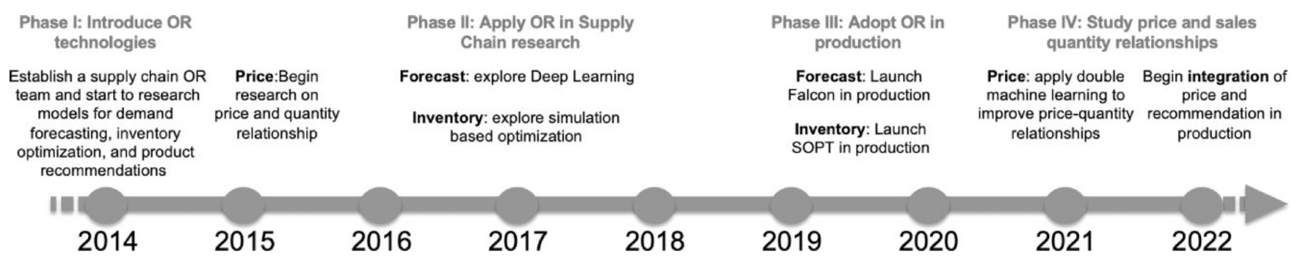
Phase III: Between 2019 and 2020, we launched Falcon, the deep-learning forecasting algorithm, and SOPT, the simulation-based optimization approach, in production resulting in superior results when compared with existing solutions.

Phase IV: In 2020, we started applying DML and various other approaches to study price and demand relationships. We adopted our models for integrated markdown optimization and product recommendations in production in 2021.

Implementation

We deployed all the algorithms we describe in this paper in Alibaba's cloud systems, which are invoked by or embedded in the various downstream supply chain planning applications through application programming

Figure 13. Timeline of Supply Chain Algorithm Development from 2014 to 2022 Outlines Alibaba's Operations Research History



interfaces (APIs) and corresponding services. Alibaba has implemented the forecasting algorithms in more than 20 of its business subsidiaries, inventory algorithms in more than 10 business subsidiaries (not all of Alibaba's businesses carry inventory), and online markdown optimization in more than six business subsidiaries. In the process, we have been careful to allow business users to be able to visualize the various factors that affect forecasts and the simulation results of optimal inventory settings, which has significantly facilitated the acceptance of these algorithms in practice. It is our belief that algorithms should not be sealed in black boxes but should be conceptually understood by business users, which is critical to the algorithms' acceptance.

Financial Benefits

Alibaba has implemented these algorithms in almost all its retail businesses over the last three years since 2019 and has generated significant financial benefits in three areas: (1) shrinkage cost and inventory holding cost reduction, (2) increased gross merchandise value (GMV), and (3) increased profit. Table 3 presents the detailed annual financial benefits across major business subsidiaries since 2019. In calculating inventory holding cost, we used a rate of return, typically between 15% and 18%, as specified by Alibaba's financial analyst. We conducted a detailed regression study on the out-of-stock decrease (service level improvement) and online conversion rate increases for each category, which served as the basis for us to translate out-of-stock decreases to gross merchandise volume or sales increases.

We estimate that these algorithms have helped Alibaba's retail business to achieve an annual $42 million cost reduction, a $110 million GMV or sales increase, and a $13 million profit increase across all business subsidiaries using our solutions. In the following, we present the succinct characteristic of each business subsidiary, algorithm implementation, and the associated financial results.

Freshippo. For Freshippo, Alibaba's omnichannel grocery retailer, we used the spatial-temporal deep learning forecast to forecast the demand of each product at each store, the SOPT inventory management algorithm to model the specific inventory consumption patterns at each store, and integrated markdown pricing and product recommendation optimization. Nearly 50% of Freshippo's sales come from its grocery sections, such as fresh produce, meat, and seafood; as such, shrinkage reduction and out-of-stock levels are critical to the business. In addition, the adoption and automation of these algorithms have eliminated a substantial amount of manual labor in order placement and price markdowns and thus dramatically reduced the number of inventory placement personnel required.

Table 4 presents the average percentage of shrinkage and out-of-stock levels and the inventory placement

Table 3. Algorithms Have Helped Alibaba to Realize Millions of Dollars of Financial Benefits Annually Since 2019

	Cost reduction (millions of dollars)		Gross merchandise volume increase (millions of dollars)		Profit increase (millions of dollars)
Business subsidiary	Shrinkage cost reduction	Inventory holding cost reduction	Increase from service level improvement	Increase from integrated markdown pricing, and product recommendation	Profit increase from promotional pricing optimization
Freshippo	20		25		
TMall Mart		5	20	35	10
Tmall Global		8		19	2.5
Others		8	5	5.5	0.5
Subtotal	20	22	37	67	13
Total	42		110		13

Note. The other subsidiaries include Lazada, Alibaba's e-business subsidiary in southeast Asia, AliHealth, Alibaba's pharmacy platform in China, and Ele.me, Alibaba's catering service platform, to name a few.

Table 4. Shrinkage and Out-of-Stock Rates and Number of Inventory Personnel Required at Freshippo Show a Dramatic Reduction After Solution Implementation

	Shrinkage rate	Out-of-stock rate	Number of inventory personnel
August 2018-July 2019	4.5%	4.8%	~200
August 2020-July 2021	3.7%	3.8%	~70

personnel required over two periods, August 2018 to July 2019 (prior to the solution implementation) and August 2020 to July 2021 (after the solution implementation).

As Table 4 shows, our algorithms reduced the shrinkage rate by 0.8%, from 4.5% to 3.7% and the out-of-stock rate by 1% from 4.8% to 3.8%. For Freshippo, these algorithms reduced the shrinkage percentage by 0.4% annually, which contributed to a $20 million shrinkage reduction and reduced the out-of-stock percentage by 0.5% annually, thus contributing to a $25 million GMV increase. In addition, Freshippo automated the algorithms and reduced inventory personnel from around 200 in 2020 to around 70 in 2022, resulting in an annual $5 million reduction in labor costs. Previously, every store employed inventory operators, who reviewed and adjusted the forecast, changed order quantities, and adjusted price markdowns based on rules of thumb. These functions were gradually consolidated at the division level and then at the corporate office, with the adoption and automation of our algorithms. Currently, 95% of Alibaba's products go through the automated forecast, inventory calculation, order placement, and markdown processes, thus freeing the inventory personnel to focus more on tasks such as seasonal events and new product introductions.

TMall Mart. TMall Mart, Alibaba's e-commerce retailer, operates a multiechelon supply chain network. We used the deep learning–based algorithm to forecast demand for this retailer and used the SOPT inventory management algorithms to solve its multiechelon inventory problems, where the vendor lead times and fill rates (i.e., the percentage of orders fulfilled) might depend on the order calendars and quantities ordered.

In the last three years (i.e., 2019–2022), these algorithms resulted in 1% increase in service levels and 10% reduction in inventory, which contributed to $5 million in cost reductions annually and a $20 million increase in GMVs from the recovery of lost sales. This again is based on the regression analysis between out-of-stock and conversion rates by category, and the coefficient is used to translate out-of-stock reductions to sales increases or lost-sales recovery.

We conducted multiple online AB experiments to evaluate the financial benefits associated with price optimization. These experiments showed that for promotional pricing of multiple products, revenue increases as high as 7% could be obtained without loss of profit margins. Overall, the multiproduct promotional pricing solutions have generated a total profit increase of $10 million annually, and the markdown optimization integrated with product recommendations generated $35 million in GMV or sales annually. These financial benefits resulted mainly from more accurate price-demand relationships and automated (i.e., set by the algorithms) promotion prices versus prices set manually (i.e., set by business personnel).

TMall Global and Others. The financial benefits at TMall Global and other subsidiaries such as Lazada are similar to those of TMall Mart and thus we do not elaborate on them. Our approaches would be applicable in both the omnichannel retail industry and many other industries that face similar challenges for demand forecasting, inventory management, price optimization, and product recommendations. We have also used these algorithms to serve our external customers across different industries, such as consumer electronics. Our algorithms are deployed in Alibaba's cloud systems and called though APIs by external customers, which again demonstrates the transportability of the algorithms.

Summary

Alibaba has pioneered novel analytical solutions, using cutting-edge algorithms in deep learning and simulation optimization, in its retail operations for improved demand forecasting, inventory management, price optimization, and product recommendations. These solutions have improved customer service levels and generated significant financial results with reductions in shrinkage and inventory costs and increases in sales volume and profits. As the retail business evolves, we believe OR will remain essential to supporting e-commerce and retail innovation and generating positive value for our society and economy.

Appendix A. Multiproduct Price Optimization Models

In the model, the price is discretized rather than continuous. Let I be the set of products, J be the set of campaign activities, and K be the set of discrete points for the price. Let x_{ijk} be the binary decision variable to select price point k for product i in campaign activity j. The function $p_{ijk}(x_{ijk})$ is the price corresponding to price point k, selected by x_{ijk}, whereas the function $q_{ijk}(x_{ijk})$ is the corresponding predicted sales.

$$\text{Max} \quad \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} p_{ijk}(x_{ijk})q_{ijk}(x_{ijk})x_{ijk} \tag{A.1}$$

Subject to :

$$\frac{\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}(p_{ijk}(x_{ijk}-c_i)q_{ijk}(x_{ijk})x_{ijk}}{\sum_{i\in 1}\sum_{j\in J}\sum_{k\in K}p_{ijk}(x_{ijk})q_{ijk}(x_{ijk})} \geq r* \tag{A.2}$$

$$\sum_{j\in J}\sum_{k\in K} x_{ijk} = 1, \ \forall i \in I \tag{A.3}$$

$$\sum_{i\in I}\sum_{k\in K} x_{ijk} \leq N_j, \ \forall j \in J \tag{A.4}$$

$$X_{ijk} \in \{0,1\}, \ \forall i \in I, j \in J, k \in K \tag{A.5}$$

Objective Function (A.1) is set to maximize the total gross merchandise value. Constraint (A.2) requires that the overall profit margin be greater or equal to r^*, the target value set by business, where c_i is the procurement price of product i. Exactly one discrete price point for each product is selected for a campaign activity, as shown in Equation (A.3). In Constraint (A.4), the number of different price points across all products for campaign activity j is required not to exceed N_j, the maximum number allowed. All the decision variables are binary, as specified in Constraint (A.5). The previous model is a nonlinear binary programming problem, and we developed a metaheuristic method, which is an evolutionary algorithm (Crepinsek et al. 2013), to improve the computational performance.

Appendix B. Price Optimization in Product Recommendation

Let K be the set of potential products indexed by k, and T be the set of potential users indexed by t. In each user session, let $S_t \leq K$ be the set of products exposed to user t. Let $ctcvr_{tk}$, ctr_{tk}, and ctr^* stand for the consumer conversion rate, click-through rate, and click-through-rate target, respectively. Let p_k and s_k be the price and salvage values of product k, respectively, and I_k be the inventory available. The product ranking optimization problem can be formulated as follows.

$$\text{Maximize} \quad \sum_{t\in T}\sum_{k\in K} ctcvr_{tk}p_k x_{tk} - \sum_{k\in K}\left(I_k - \sum_{t\in T} ctcvr_{tk}x_{tk}\right)s_k \tag{B.1}$$

Subject to :

$$\sum_{t\in T} ctcvr_{tk}x_{tk} \leq I_k, \ \forall k \in K \tag{B.2}$$

$$\sum_{t\in T}\sum_{k\in K} ctcvr_{tk}x_{tk} \geq ctr* \tag{B.3}$$

$$\sum_{k\in K} x_{tk} \leq S_t, \ \ \forall \ t \in T \tag{B.4}$$

$$x_{tk} \in \{0,1\}, \ \forall t \in T, k \in K \tag{B.5}$$

The objective is to find the optimal assignment of product k to user t, x_{tk}, which maximizes the total revenue, as the first term in Objective Function (B.1), but with a penalty for salvage if the number of units sold is less than the available inventory level, as given in the second term in Objective Function (B.1). Constraint (B.2) requires that units sold should not exceed the available inventory. Constraint (B.3) guarantees that the overall click-through rate (CTR) for each product across all users should be higher than the target CTR ctr^*. Constraint (B.4) ensures that a limited number of products is assigned to user t. The decision variables, x_{tk}, are binary, as specified in Constraint (B.5). A typical problem of a five-minute timeframe has a size of 60,000 customers (T = 60,000) and 10,000 products (K = 10,000), which cannot be solved directly using existing solvers. Thus, we devised an online matching Dantzig-Wolfe and Dual Descent Network algorithm to obtain high-quality online solutions, within 99.8% of optimality on average, returning a solution in five minutes.

Let α_k denote the dual prices of Constraint (B.2). We then use the following decision rule in Step 3:

$$x_{tk} = \begin{Bmatrix} 1, \ k \in \{top\ S\ of\ ctcvr_{tk}(p_k + s_k) - \sum_k \alpha_k ctcvr_{tk} \\ 0,\ otherwise, \end{Bmatrix}$$

where the "*top S*" returns the S largest elements of the given input set (i.e., the candidate set of products to recommend to customers), ranking the selected S products in descending order of $ctcvr_{tk}(p_k + s_k) - \sum_k \alpha_k ctcvr_{tk}$.

References

Agrawal S, Wang Z, Ye Y (2014) A dynamic near-optimal algorithm for online linear programming. *Oper. Res.* 62(4):876–890.

Alibaba (2022) 2022 Edelman finalist: Alibaba. Accessed August 26, 2022, https://www.youtube.com/watch?v=80opt8vDkd8.

Chen H, Dai B, Li Y, Zhang Y, Wang X, Deng Y (2022) Stock allocation in a two-echelon distribution system is controlled by (s, S) policies. *Internat. J. Production Res.* 60(3):894–911.

Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C, Newey W, Robins J (2018) Double/debiased machine learning for treatment and causal parameters. *Econom. J.* 21(1):C1–C68.

Crepinsek M, Liu S, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Survey* 45(3):1–33.

Feldman J, Zhang D, Liu X, Zhang N (2022) Customer choice models vs. machine learning: Finding optimal product displays on Alibaba. *Oper. Res.* 70(1):309–328.

Fu M, Healy K (1997) Techniques for optimization via simulation: An experimental study on an (s, S) inventory system. *IIE Trans.* 29(3):191–199.

Gijsbrechts J, Boute R, Van Mieghem J, Zhang D (2022) Can deep reinforcement learning improve inventory management? Performance on lost sales, dual sourcing, and multi-echelon problems. *Manufacturing Service Oper. Management* 24(3):1349–1368.

Haijema R, van der Wal J, van Dijk N (2007) Blood platelet production: Optimization by dynamic programming and simulation. *Comput. Oper. Res.* 34(3):760–779.

Harsha P, Subramanian S, Uichanco J (2019) Dynamic pricing of omnichannel inventories. *Manufacturing Service Oper. Management* 21(1):47–65.

Hu H, Zhang Y, Wei J, Zhan Y, Zhang X, Huang S, Ma G, et al. (2022) Alibaba VRP algorithms enable rapid pick and delivery. *INFORMS J. Appl. Anal.* 52(1):27–41.

Oreshkin B, Carpov D, Chapados N, Bengio Y (2019) N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. Accessed July 20, 2022, https://arxiv.org/pdf/1905.10437v1.pdf.

Perakis G, Sood A (2006) Competitive multi-period pricing for perishable products: A robust optimization approach. *Math. Programming* 107(1–2):295–335.

Salinas D, Flunkert V, Gasthaus J, Januschowask T (2020) DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Internat. J. Forecasting* 36(3):1181–1191.

Taylor S, Letham B (2018) Forecasting at scale. *Amer. Statist.* 72(1):37–45.

van de Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N (2016) WaveNet: A generative model for raw audio. Accessed July 20, 2022, https://arxiv.org/pdf/1609.03499.pdf.

Wan X (2022) Omnichannel distribution to fulfill retail orders. *Manufacturing Service Oper. Management* 24(4):2150–2165.

Zhang X, Meiser D, Liu Y, Bonner B, Lin L (2014) Kroger uses simulation-optimization to improve pharmacy inventory management. *J. Appl. Anal.* 44(1):70–84.

Yuming Deng is a director in Alibaba Group, leading the Digital Supply Chain department. He was a finalist for the 2021 Franz Edelman Award for Achievement in Operations Research and the Management Sciences by INFORMS for "Alibaba Vehicle Routing Algorithms Enable Rapid Pick and Delivery." His research interests are assortment planning, network planning, pricing strategy, forecasting, inventory optimization, and simulation-based optimization and its applications.

Xinhui Zhang is a senior director and head of supply chain and operations research committee at Alibaba Group. He was a finalist for the 2013 and 2021 Franz Edelman Award for Achievement in Operations Research and the Management Sciences by INFORMS. His research interests include large-scale optimization, inventory management, vehicle routing, and various aspects of supply chain management.

Tong Wang is a director at Alibaba Group, leading the Supply Chain Forecasting Team. Prior to Alibaba, he was a tenured associate professor at National University of Singapore. His research is on information and flexibility in supply chain management, with a special focus on retail analytics. He obtained his PhD in decision sciences from INSEAD at France.

Lin Wang is a senior algorithm expert at the Alibaba Digital Supply Chain Group. His recent research and application interest are particularly focused on forecasting, using state-of-the-art optimization, deep learning, and automated machine learning techniques. He obtained his MS degree from the School of Computer Science and Engineering at Beihang University.

Yidong Zhang is a director at Alibaba Group, leading the B2C Supply Chain Planning and Inbound Planning Optimization team. Prior to Alibaba, he was a research scientist at BASF, conducting projects to optimize the supply chain effiency within BASF globally. He obtained his PhD from the Industrial and Systems Engineering Department at University at Buffalo.

Xiaoqing Wang is a director at Alibaba Group, leading the Supply Chain, Fulfillment and Service Optimization. His research interests include the combination of operation research, stochastic optimization, and large-scale online optimization in the areas of logistics and retail industry. He obtained his PhD degree in systems engineering at Northeastern University.

Su Zhao is a staff algorithm engineer at Alibaba Group, focusing on pricing and revenue management. Prior to Alibaba, she was a senior algorithm engineer at AInnovation and a quantitative researcher at Jinde Hedge Fund. She obtained her PhD from the Industrial and Systems Engineering Department at Texas A&M University and MS and BS degrees in control science and engineering at Tsinghua University.

Yunwei Qi is a senior algorithm advisor at Alibaba Group, leading the pricing group and B2C supply chain team for Tmall Supermarket. His research interests include recommendations systems, forecasting, revenue management, causal inference, large-scale optimization, and stochastic optimization. He obtained his PhD in industrial systems and engineering at The Ohio State University.

Guangyao Yang is a senior algorithm expert of Alibaba Freshippo Group, leading the Freshippo supply chain and revenue management team. His research is on forecasting, inventory, pricing, and promotion strategy, with a special focus on the application of spatio-temporal graph convolutional neural networks in the supply chain. Before that, he was an algorithm engineer at JD.com.

Xuezheng Peng is a senior algorithm expert at Alibaba Freshippo Group, leading the recommendation systems group. His research interests include artificial intelligence and human-computer interactions, including personalized recommendation system, search engine, dialogue system, knowledge graph, and business applications. He obtained his MS degree from the School of Electronics Engineering and Computer Science at Peking University.