IKEA: Innovative Know-how for E-commerce Allocation

Yanfei Chen[1] Mathijs Pellemans[2] Francisco Richter[3]
Yasmin Roshandel [2] Vincent Schmeits[4] Peter Waldert [5]
Laith Wehbi[2] Tom C. van der Zanden[6]

Abstract

We examine the optimization of order fulfillment strategies employed by IKEA for orders placed via its online webshop. IKEA currently utilizes a strategy whereby orders are assigned to local stores or warehouses based on the minimization of picking and delivery costs without consideration for the potential impact of future orders. This approach often leads to the early exhaustion of order picking capacity in several stores, resulting in increased shipping costs and, in some cases, the inability to fulfill orders as orders received later in the day are assigned to more distant stores. We propose alternative strategies aimed at enhancing the robustness of IKEA's order allocation process against incoming future orders. Our solutions include a heuristic approach that introduces a penalty for selecting warehouses with low residual capacity, a linear programming model that integrates predictions of future orders, and a neural network that is trained on optimal (offline) solutions derived from historical order allocation data. We show that these proposed methods can significantly reduce both the total fulfillment costs and the proportion of orders that cannot be fulfilled, offering a more efficient and reliable order fulfillment system for IKEA.

Keywords: E-commerce order fulfillment optimization, Predictive analytics in supply chain, Capacity planning and management.

[1] Research Centre for Operations Management, KU Leuven, Belgium.
[2] Department of Mathematics, VU Amsterdam, The Netherlands.
[3] Institute of Computing, Università della Svizzera Italiana, Switzerland.
[4] Korteweg-de Vries Institute for Mathematics, University of Amsterdam, The Netherlands.
[5] Technical University of Graz, Austria.
[6] Department of Data Analytics and Digitalisation, Maastricht University, The Netherlands.

1 Introduction

1.1 Problem Background

Large e-commerce retailers often have multiple sites from which orders can be fulfilled. As orders come in, decisions have to be made as to what order to fulfill from what site. Due to limited warehouse capacity, decisions made in the past can affect what options are available for fulfilling future orders. In this paper, we study a related problem put forward by IKEA at SWI: how should orders received through IKEA's webshop be allocated for fulfillment to its stores, taking into account potential future orders?

This problem requires balancing two factors: on the one hand, shipping costs are affected by the geographical proximity of the customer to the fulfilment site. Ideally, we would like each order to be fulfilled by the site, which results in the lowest cost for that order. This is the approach currently used by IKEA: as orders come in through their webshop, they immediately allocate each order to the warehouse that can handle it most economically.

However, as the sites may have limited capacities (e.g., limited stock or limited picking capacity), such a greedy strategy might cause the capacity of a warehouse to become exhausted, resulting in higher costs for orders received later. This means that we have to take into account not only immediate fulfillment costs but also the remaining capacity of each warehouse and a prediction of future orders. For example, if an order is received that can be fulfilled by two warehouses at roughly equal cost, it might make sense to allocate it to the warehouse whose capacity is least likely to be exhausted. Otherwise, we might incur high costs for shipping future orders that are close to one warehouse but farther away from the other. In this paper, we propose several methods for determining how fulfillment costs and preservation of remaining capacity can be balanced.

The significance of reducing logistics costs extends beyond mere financial savings; in fact, low costs and sustainability are closely related (Beliën et al., 2017). Reducing reliance on transportation translates to lower emissions of greenhouse gases (GHGs), which is one of the crucial targets of Sustainable Development Goal (SDG) 13: Climate Action (United Nations, 2024). Particularly in Europe today, transport emissions account for approximately a quarter of the EU's total GHG emissions (European Environmental Agency, 2024). These considerations underscore the importance of the problem addressed in this paper.

1.2 Research Questions

In this study, our aim is to develop a strategy to allocate each online order to a fulfillment center in such a way that daily total costs are minimized and the number of fulfilled orders is maximized. To guide our research, we establish the following research questions based on IKEA's stated interest:

RQ1 How do different allocation strategies impact both the feasibility and costs of order fulfillment?

RQ2 What is the potential for optimization in order allocation when all orders are known in advance?

RQ3 How does a batch processing approach, where orders are evaluated for feasibility before allocation, compare to immediate allocation strategies in terms of feasibility and cost efficiency?

RQ4 What are the implications of enabling order splitting on the feasibility and cost of order allocation strategies, and how do these insights differ from strategies without order splitting?

RQ5 Which configuration parameters, such as available resources and picking costs, serve as primary bottlenecks in achieving complete fulfillment of orders, and how do adjustments to these parameters impact overall fulfillment rates and costs?

1.3 Related Literature

This study primarily contributes to the literature on online order fulfillment and the stochastic generalized assignment problem. Order fulfillment has been extensively studied by numerous researchers over the decades, and we refer to Croxton (2003) and Ricker and Kalakota (1999) for a comprehensive review of this research area. The rapid growth of e-commerce has led to a paradigm shift in order fulfillment methodologies. As firms face the challenges of efficiently managing multi-echelon distribution networks, numerous studies have underscored the significance of integrating inventory optimization and order allocation. We cite Ishfaq and Bajwa (2019) who introduce a non-linear mixed-integer profit maximization model of the online order fulfillment process for multi-channel retailers, Zhao et al. (2022) who provide theoretical bounds and empirical validation for a myopic fulfillment policy in e-commerce logistics, Levin (2023) who introduces a real-time control policy for online order fulfillment that improves throughput and customer service under uncertainty, Rao et al. (2011) who empirically show that order fulfillment delays in online retailing negatively affect customer shopping behavior, and Onal et al. (2023) who introduce efficient heuristics for improving picking efficiency in fulfillment warehouses.

The problem that we address in this study is similar to the *stochastic generalized assignment problem* (SGAP) (Albareda-Sambola et al., 2006), which is an extension of the *generalized assignment problem* (GAP). The GAP itself is an extension of the traditional, well-studied assignment problem. The assignment problem addresses situations where a number of items need to be assigned to a number of agents, e.g., assigning workers to tasks or programs to computers. Algorithms like the *Hungarian method* have been developed to solve assignment problems efficiently (Conforti et al., 2014, p. 149). If each agent has a certain capacity, which is found in many real-world situations, the classical assignment problem becomes the GAP. For a comprehensive survey of the traditional GAP, we refer to the work by Öncan (2007).

In reality, uncertainties may reduce the validity of some GAP models. Possible situations include a change in agent capacity, no-show of agents, and so on. These concerns motivate the incorporation of uncertainties into the GAP, hence the study of SGAP. There is not a lot of rich literature available on SGAP (Albareda-Sambola et al., 2006), yet we cite Alaei et al. (2013) who introduce a $1 - \frac{1}{\sqrt{k}}$-competitive algorithm for online SGAPs, Albareda-Sambola et al. (2006) who allow reassignments to be performed if there are overloaded agents, Li et al. (2023) who introduce an algorithm for the online SGAP with unknown Poisson arrivals, Morton et al. (2009) who develop a branch-and-price approach to the SGAP, Sarin et al. (2014) who develop a branch-and-price approach for the stochastic generalized assignment machine scheduling problem, and Spoerl and Wood (2003) who develop a stochastic version of the *elastic generalized assignment problem* (EGAP).

However, it should be noted that, in an e-commerce environment, order allocation does not need to be determined instantly, and sometimes, 'A little delay is all we need' (Xie et al., 2023). Although delay is not an option, as IKEA has to assign the customer to a fulfillment center as soon as the order is placed, we still discuss the general ideas to inspire future operational processes. Recently, Xie et al. (2023) explore the benefits of delaying real-time decisions, drawing on research by Zhao et al. (2022) that indicates minor delays in order processing improve outcomes in dual-layer networks, a concept also observed by Wang et al. (2023) in practices involving short delays. Xie et al. (2023) demonstrate that in online decision-making, the difference in performance between strategies with delays and optimal offline strategies narrows exponentially with increased delay duration. Their research, involving extensive numerical experiments with both synthetic and real-world data, highlights that minimal delays are most advantageous, with longer delays offering diminishing returns. Additionally, in contexts where demand distribution is unknown, delays are shown to be beneficial for algorithmic learning. Xie et al. (2023) also investigate batch decision-making, finding it to be a less effective form of delay compared to individual delays.

1.4 Paper Outline

Our paper is structured as follows. The dataset provided by IKEA is explored in Section 2. We provide the mathematical formulation and the optimal offline solution in **??**. Our strategies for the problem are detailed in Section 3. Computational results and their evaluations are documented in Section 4, including a discussion on the implications of the results. The final section is reserved for conclusions.

2 Exploratory Data Analysis

The dataset provided by IKEA provides a representative sample of their online ordering process, including a variety of variables such as order frequency, item weight, and availability, which are important for understanding the dynamics of order fulfillment. In our exploratory data analysis (EDA), we created visualizations to uncover patterns and insights to guide our optimization strategies in the subsequent sections.

Daily Number of Orders. We analyze order frequency by investigating the distribution of orders over time.

Figure 1 shows a fluctuating pattern of daily orders over the course of a month. These significant fluctuations are potentially caused by factors such as weekly cycles, holidays, and promotional activities. There is a sharp increase towards the end of the period after a dip, possibly corresponding to a holiday period or resulting from a marketing campaign.

Order Forecast. Forecasting the actual number of orders that will arrive during a day can be a challenging task. In principle, robust time series models or machine learning models are suitable for use in such cases. However, having enough historical data is a vital requirement for accurately training such models. Our dataset only consists of one month of orders, which likely is not enough to obtain a sufficient level of accuracy.

Despite this limitation, we deployed two time series models to assess their potential application. The goal of both models is to predict the number of daily orders. Using the 31-day dataset, we trained the models on the first 21 days and tested them on the last 10 days. Seasonal Autoregressive Integrated Moving-Average

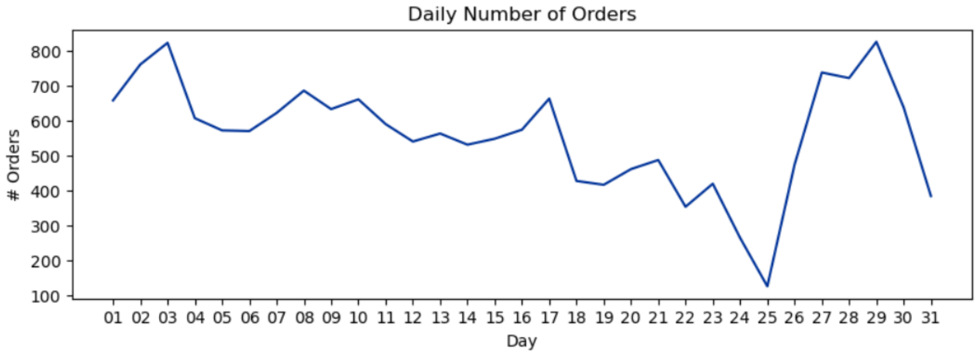


Figure 1: Daily number of online orders at IKEA over the course of a month.

with Exogenous Regressors (SARIMAX) has shown capability to capture patterns for small datasets (Tarsitano and Amerise, 2017), while TimeGPT has demonstrated promising results to produce accurate forecasts across various domains (Garza and Mergenthaler-Canseco, 2023).

Figure 2 shows the observed data, test data, and predicted daily orders. The predictions follow the trend but cannot capture dips and peaks in the data which corresponds to the initial hypothesis that such models would likely not give accurate forecasts with such a dataset size.

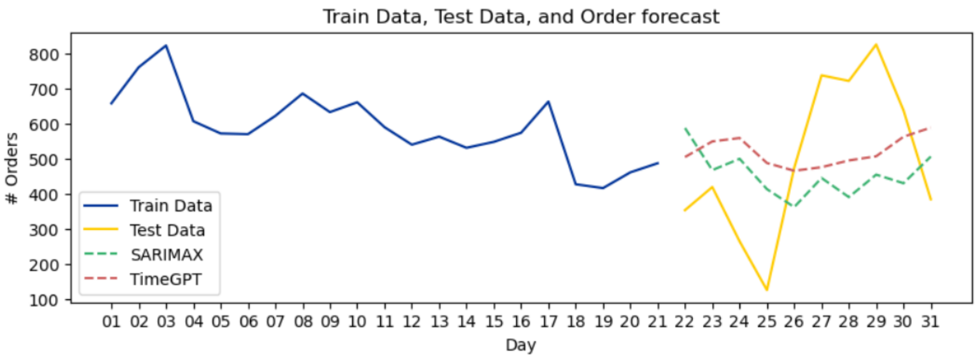


Figure 2: Order forecasts deploying two time series models: SARIMAX and TimeGPT.

Furthermore, we deployed additive time series decomposition to test for seasonality influences. The observed number of orders $Y(t)$ is decomposed into trend, seasonality, and residual components in Equation (1). The day of the week is found to be not statistically significant in influencing the number of orders in the 30-day data set, as seen in Figure 3.

$$Y(t) = T(t) + S(t) + R(t). \tag{1}$$

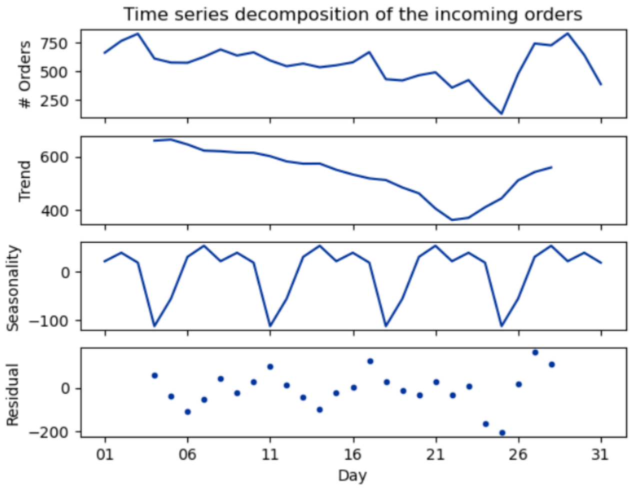


Figure 3: Time series decomposition of the incoming number of orders. There seems to be a weekly seasonal pattern, however it was not statistically significant. Kruskall Wallis test (p-value=0.47).

Item Frequency. An overview of items that constitutionally account for a large portion of the orders can be seen in Figure 4. The graph follows a Pareto distribution, suggesting that a small number of items constitute a significant portion of the total number of orders. A statistical Kolmogorov-Smirnov test (Berger and Zhou, 2014) was performed to evaluate the fit of the item frequency distribution to a Pareto distribution, resulting in a p-value of approximately 0.143. This result indicates that there is not enough statistical evidence to reject the starting hypothesis that the distribution follows a Pareto distribution, confirming that a small number of items significantly influence the total order volume. The first item appears to be an outlier, dominating the frequency chart with a significantly higher order count.

The percentage contribution line overlaid on the bar graph decreases almost linearly, indicating that each item subsequently contributes less to the overall order volume. This pattern underscores the importance of focusing on inventory and supply chain strategies for these top-selling items, given their potential to impact customer satisfaction and order fulfillment efficiency. It also suggests that good inventory management of these items is crucial, as they drive a substantial portion of the business.

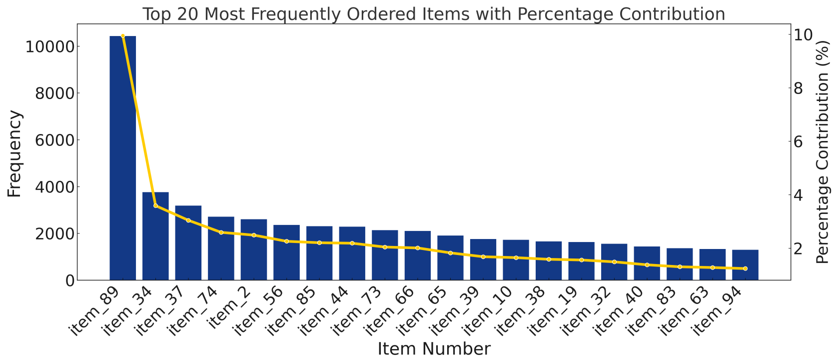


Figure 4: Top 20 Most Frequently Ordered Items.

Item Availability. As the items described in the previous paragraph represent a large portion of the orders, it is important to examine their availability across the various IKEA stores. Figure 5 provides a visual representation of the availability of these high-demand items across stores. The rows correspond to the top twenty items that were ordered. The columns represent individual stores. The majority of blue across most items and stores suggests a good alignment between IKEA's inventory strategy and its demand patterns. However, there are noticeable gaps in availability for some items in specific stores.

Item Weights and Order Complexity. Another important aspect of the analysis is examining the behavior of item weights and complexity in orders. Figure 6 shows the distribution of both the total weight of orders and the distribution of item weights in orders. Both show a highly left-skewed distribution, and these plots indicate that most orders are light, with a few orders having significantly higher total weight. This shows that IKEA's order profile is mainly composed of smaller, lighter items, with occasional larger purchases.

The complexity distribution of the orders, i.e., the number of unique items in each order, is shown in Figure 7. We find that the majority of orders are relatively simple, containing a few unique items. As the number of unique items per order increases, the frequency of such orders sharply declines, indicating that orders with

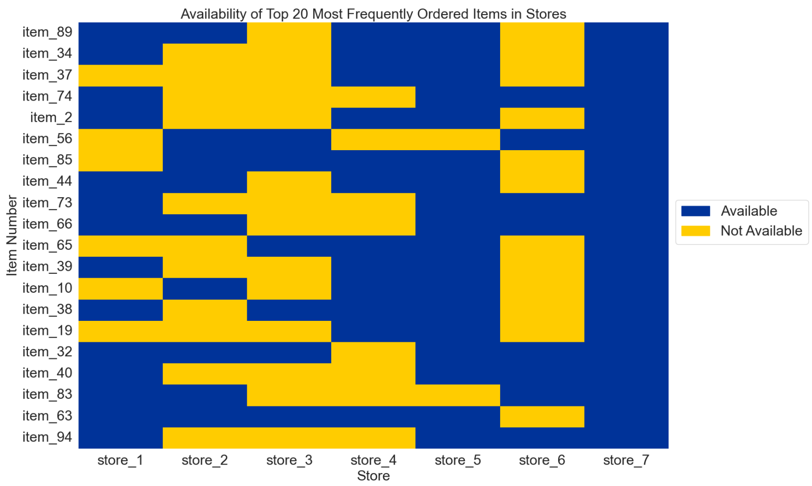


Figure 5: Availability heatmap for the top 20 most frequently ordered items.

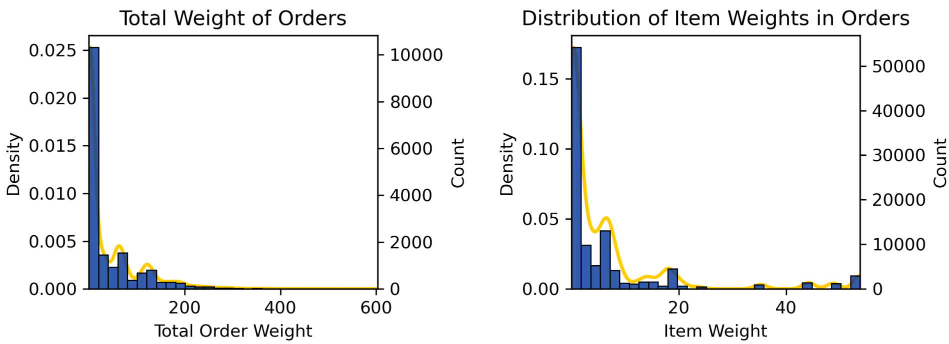


Figure 6: Distribution of total order weights.

higher complexity are considerably less common. This trend continues, with orders containing more than five unique items becoming increasingly rare.

Correlations. To investigate the correlation among item weight, frequency, order date, and store availability, we constructed the correlation heatmap displayed in Figure 8.

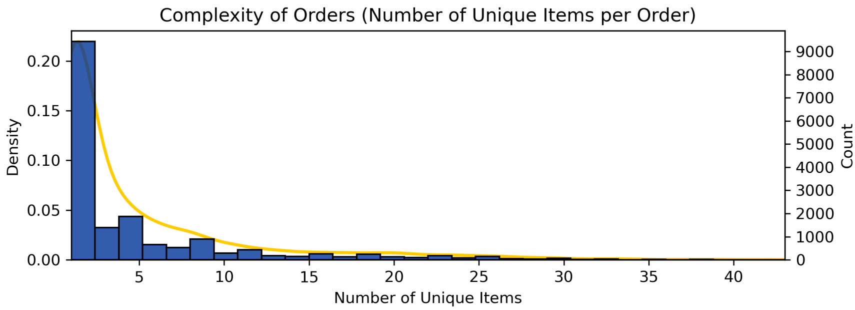


Figure 7: Distribution of order complexity.

There is a strong negative correlation of −0.88 between *item weight* and *store_5* availability, suggesting that heavier items are less likely to be found in *store_5*. Contrarily, *store_6* shows a strong positive correlation of 0.64 with *item weight*, indicating that this store tends to stock heavier items more frequently.

The availability of items in *store 1* and *store 3* shows a moderate positive correlation with *item frequency*, with correlation coefficients of 0.22 and 0.15, respectively. This implies that more frequently ordered items are more likely to be available in these stores.

Store_4 shows an interesting pattern, with a perfect negative correlation of −1.00 with *store_6*, indicating a mutually exclusive stocking policy between these two stores. This pattern could be due to a variety of reasons, such as store specialization, geographical considerations, or customer demographic differences.

Cluster Analysis. Given the results of the order inter-dependencies and correlations, we can deduce that clustering items across orders could be beneficial, especially when IKEA considers order splitting, which could be based on different item characteristics. We, therefore, deployed two clustering algorithms to create clusters of similar items based on individual weight and frequency found across all orders: K-means (Jin and Han, 2010) and Density-Based Spatial Clustering of Applications with Noise DBSCAN (Ester et al., 1996).

Figure 9 shows the application of the K-means and DBSCAN clustering algorithm to the orders dataset, scaled based on item weight and frequency. The elbow method plot on the top left identifies the optimal number of clusters, with the *elbow point* occurring at three clusters. This suggests that in the data, we can distinguish three groupings of items that offer the most significant reduction in within-cluster variance without unnecessarily increasing the complexity of the model. The *Scaled*

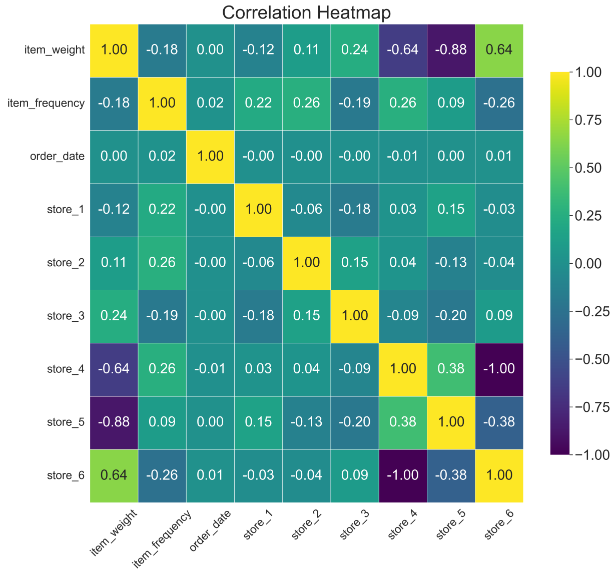


Figure 8: Correlation Heatmap among *item weight*, *item frequency*, *order date*, and the available stores (fulfilment centres).

K-means Clustering plot on the right visualizes these three clusters. Each point represents an item, plotted by its scaled weight and frequency. The clustering reveals different densities and distributions: a slightly dense cluster near the origin, a moderately dispersed cluster above it, and a more sparse cluster further along the axis of scaled item weight.

The two bottom plots in Figure 9 present the clustering results of the K-means and DBSCAN ($\epsilon = 0.5$) algorithms for items based on unscaled weight and frequency. The K-means algorithm, with the number of clusters set to three, identifies groups with distinct characteristics:

- Cluster 0 is comprised of items with a low average of item weights of 1.82 and a high average frequency of 2129.50. This cluster represents the light items that are more frequently ordered, possibly indicating high-demand products.

- Cluster 1, the most dense with over 81 items, has a moderate mean item weight of 4.97 and a low to moderate item frequency mean of 667.30. This cluster represents the middle ground between clusters 0 and 2, where it might contain items not as light as those in cluster 0 but still have a decent level of demand,

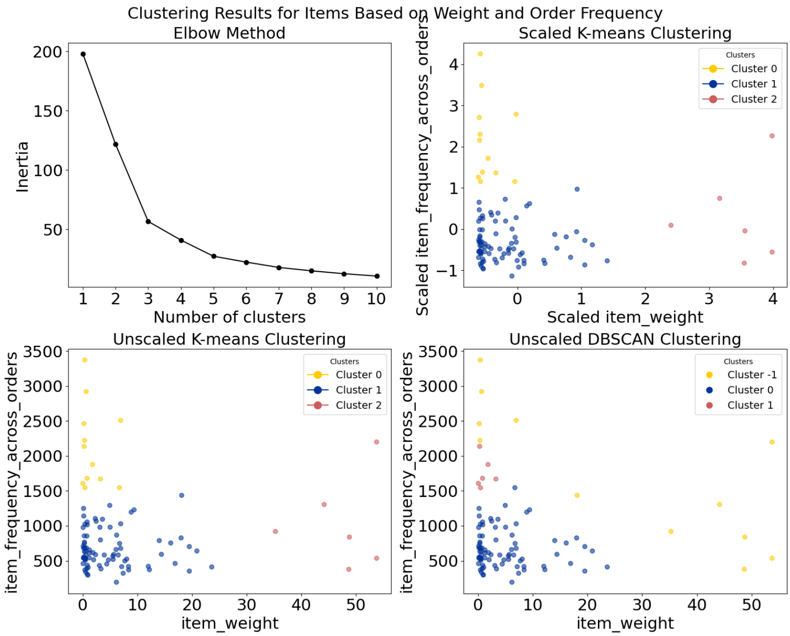


Figure 9: K-means and DBSCAN results for clustering based on item weight and frequency across all orders.

although not as high.

- Cluster 2 consists of orders with a high mean item weight of 47.39 and a mean item frequency of 1029.50, possibly representing the bulkier and more frequently ordered products.

The DBSCAN algorithm, known for its ability to find arbitrarily shaped clusters and identify outliers, presents a different view:

- Cluster -1, containing 12 items with a high mean item weight of 25.91 and frequency of 1758.42, indicating a small group of heavy and frequently ordered items. Cluster -1 in the output of DBSCAN typically represents a segment of outlier points.
- Cluster 0 is the most substantial, with over 81 items, showcasing a low to moderate mean item weight of 4.83 and frequency of 668.65.
- Cluster 1 contains the smallest number of items, with a low mean item weight of 1.09, but a high frequency of 1752.83 across all orders. This suggests that

these few items might be standard purchases of high-demand small products.

The various insights gained from the EDA provide us with a comprehensive understanding of the problem dynamics and IKEA's order allocation strategy. Such insights drive the subsequent strategies that have the potential to contribute to effective and efficient order fulfillment.

3 Order Allocation Strategies

In this section, we discuss the strategies that we compare in our results in Section 4. The greedy algorithm (with and without order splitting) is a basic strategy that has been provided by IKEA.

3.1 Greedy Algorithm

A simple yet effective approach is to use a greedy algorithm that assigns each incoming order to the facility that minimizes the cost function immediately. We use this model as a baseline to compare all strategies. For each subsequent order, the greedy algorithm chooses the store that minimizes the cost of fulfillment over all stores u that are able to fulfill order k, i.e., the individual terms in the sum of Function (**??**). Thus, the objective function to minimize becomes:

$$C_k\left(y^k\right) := \sum_{u\in[U_k]} \left(C_u^{FF} W_k^{tot} + C^T D_{z_k,u}\right) y_u^k, \tag{2}$$

where $[U_k]$ is the set of all stores that have sufficient remaining capacity to fulfill order y^k after the allocation of all $k-1$ previous orders. Note that all constraints (**??**)-(**??**) in the ILP formulation of the optimal solution must still be adhered to. As we allocate each order to a store directly upon arrival, the previously assigned $k-1$ orders may significantly affect the possibilities when allocating any future order k. This limitation is included in $C_k\left(y^k\right)$ by choosing between stores in $[U_k]$ instead of $[U]$.

3.1.1 Extension to order splitting

As an extension to the basic greedy model, we also considered the allowance of order splitting. If orders are allowed to be split, we instead choose the optimal pair (ϕ^k, y^k) in (**??**), again with the restriction that we can only choose from the set $[U_k]$ of all stores that have the sufficient remaining capacity to fulfill order y^k after allocating the $k-1$ previous orders. So the objective function of the extended greedy model to minimize becomes:

$$C_k(\phi^k, y^k) = \sum_{u\in[U_k]} \left(C^{SU} + C^T D_{Z_k,u}\right)\phi_u^k + \sum_{i\in I_k} C_u^{FF} W_i y_{i,u}^k. \tag{3}$$

We do not thoroughly consider order splitting beyond the greedy algorithm, and the greedy algorithm with order splitting should not be compared to the other strategies discussed in this section. Note that in order to properly compare potential strategies that do allow for order splitting, one should also extend the offline optimal solution as given in **??** to allow for splitting.

3.2 Heuristic - Greedy with Residual Capacities

Greedy attains the (offline) optimal solution on more quiet days, i.e., on days where the maximum capacity of stores is not reached with the greedy strategy (2). Whenever there is some maximum capacity reached throughout the day, the performances of the optimal solution and greedy do differ. On such days, the greedy algorithm performs worse by a substantial amount: many orders are not being fulfilled (or are assigned to a sub-optimal store) because the maximal capacities of more favorable stores are reached. In the heuristic model, we attempt to improve the greedy strategy by taking into account the remaining capacity of each store when assigning orders. We will provide a description of two versions of the heuristic model, the basic and extended model. Here, the extended model is a generalization of the basic model.

3.2.1 Basic Heuristic Model

The main issue in the greedy approach during busier days stems from the variability among stores. Certain stores may be substantially cheaper than others, for instance, stores with a low picking cost per weight or with a central position in the area. As the greedy approach favors the cheapest option for each order, such stores run out of capacity very quickly. If, at some point during the day, all stores that contain some item i have reached their maximal capacities, then every subsequent order that includes item i cannot be fulfilled, and a penalty cost will be given to these orders. To remedy this problem, we propose the following heuristic: we let the choice of a store depend not only on the cost to fulfill order k at store u, but also on the remaining capacity of store u for that day. This ensures that the orders are distributed more evenly across stores. In turn, we can fulfill more orders, leading to reduced penalty costs.

Similar to the greedy approach, we assign each order instantly without prior information about future orders. Additionally, the basic heuristic uses information from previous orders to determine the optimal store for the order. We define the *residual capacity* R_u and *normalised residual capacity* $\tilde{R}_u$ of a store u when evaluating order k as:

$$R_u := M_u^W - \sum_{j=1}^{k-1} W_j^{tot} y_u^j, \quad \text{where} \quad \tilde{R}_u := \frac{R_u}{M_u^W}. \tag{4}$$

For each order k, the basic heuristic minimizes the objective function

$$C_k\left(y^k\right) := \sum_{u \in [U_k]} \frac{1}{\tilde{R}_u} \cdot \left(C_u^{FF} W_k^{tot} + C^T D_{z_k,u}\right) y_u^k. \tag{5}$$

Again, note that we can still only choose any store $u \in [U_k]$ with sufficient remaining capacity. To summarise, compared to (2), the new formula includes a factor that weighs in the remaining available capacity for each store. Note that at the start of the day, $\tilde{R}_u = 1$ for all u, so that the greedy and heuristic approaches will make the same choice. As more orders are fulfilled during the day, if the normalized residual capacity of the store u increases relatively quickly, it will become more likely that an order gets fulfilled by another store, even if store u can fulfill it at the lowest cost. See also Figure 10. Note that, similar to the greedy algorithm, an order is always fulfilled as long as there is at least one store that is able to do so.

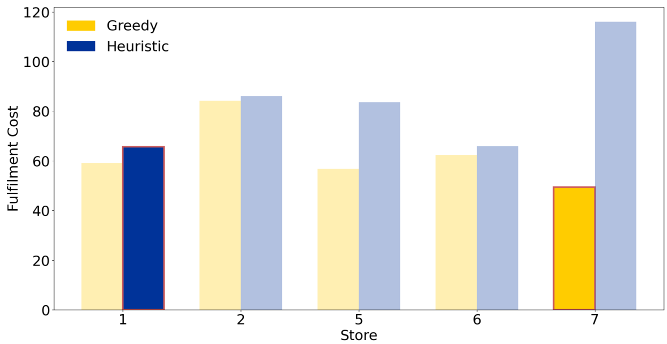


Figure 10: Values of the separate terms in objective functions (2) (greedy) and (5) (heuristic) for a specific order k. Stores 1, 2, 5, 6, and 7 are able to fulfill the order. The greedy algorithm chooses to store 7, minimizing the cost function. The heuristic algorithm weighs in the residual capacity and picks store 1.

3.2.2 Extension to Busy Day Factor

A natural question to the comparison of the basic heuristic and the greedy algorithm is: to what extent should the normalized residual capacity $\tilde{R}_u$ influence the choice of the store? On days when it is possible to fulfill every order at the lowest cost, the greedy algorithm is optimal and might outperform the basic heuristic algorithm. Conversely, on busy days, it may be more beneficial to base the choice of store solely on the normalized residual capacities and disregard the cost of fulfilling an order completely to maximize order fulfillment. We introduce the *busy day factor* $\beta \in [0, 1]$ that quantifies the influence of the normalized residual. We generalize the function in (5) to the following:

$$C_k\left(y^k\right) := \left(\frac{1}{\tilde{R}_u}\right)^{\beta} \cdot \left(C_u^{FF} W_k^{tot} + C^T D_{z_k,u}\right)^{1-\beta}. \tag{6}$$

Note that β set to 0.5 is equivalent to the basic heuristic approach, as $x \mapsto x^{\beta}$ is an increasing function and distributes over multiplication. Figure 11 illustrates how different values of β affect the extended heuristic's performance compared to greedy. In practice, one could choose the busy day factor based on a prediction for the number of orders on a specific day (with a low value on quiet days, and a higher value on busy days). Alternatively, one could use a different factor β_k for each order on the same day. For instance, by initializing β_1 to 0 and adjusting the value as the day progresses based on the observed busyness.

When β is set to 1, we completely disregard any characteristics of the order and simply pick the store that has the most relative capacity left, i.e., with the highest *normalized residual capacity* $\tilde{R}_u$. We can see in Figure 11 that this choice of β is optimal on Day 3 (which is a busy day). This can be expected based on the computation of the final cost in this model: each unfulfilled order adds a set penalty cost of C^P to the total. If C^P increases, fulfilling as many orders as possible becomes significantly more beneficial than fulfilling each individual order at the optimal store. Modifying the value of C^P, or introducing a variable penalty cost C_k^P dependent on order k, could change this outcome.

3.3 (I)LP with predictions

In **??**, we discussed how the optimal solution is obtained by solving an ILP if all orders are known in advance. Our challenge lies in allocating the current order without prior knowledge of future orders. However, if we have a prediction of expected future orders based on historical data, we could use this to solve the allocation of the current order in the same way. Plugging in the predicted orders along with the current order in the ILP in Subsecton **??**, we obtain a solution for allocating the current order.

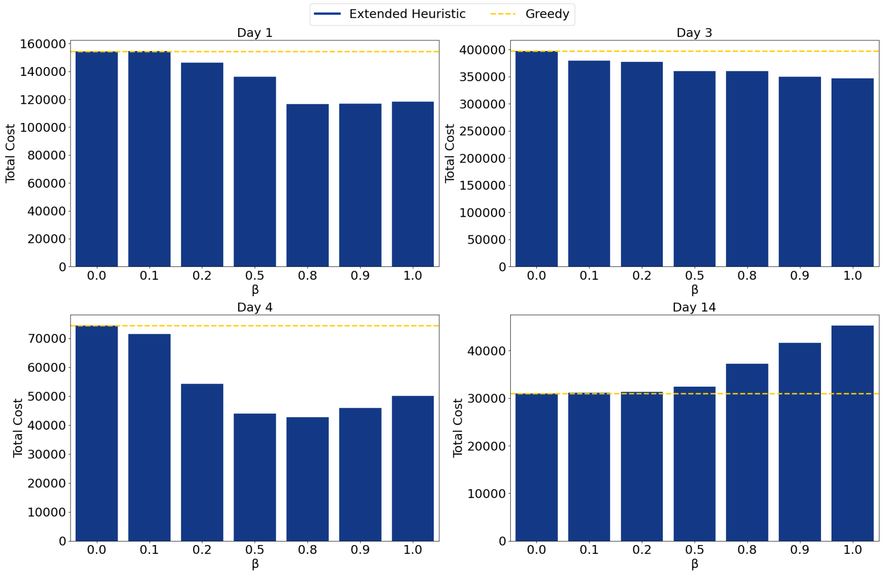


Figure 11: Comparison of the extended heuristic and greedy approach on four different days for varying values of β. On busy days 1 and 3 (above average of total orders), we find higher values of β to be optimal. Day 4 is more quiet (close to average), and the optimal value for β lies between 0.5 and 0.9. On notably quiet day 14 (below average), the greedy approach is optimal.

3.3.1 Basic (I)LP Model

We use a very simple prediction of expected orders. As the prediction of orders on a typical day, we sample 600 orders at random from orders in the last fifteen days of the orders dataset (and we always use the same, fixed, random sample). The number 600 is chosen because it is close to the average number of orders per day in the orders dataset.

Depending on our estimate of the time of day, we take a corresponding portion of those 600 orders as a way of predicting which orders were still to come. Since the dataset does not include the time of day each order was received, we assume the orders were received at an even rate throughout the day.

Note that, with this model, we would essentially execute the offline optimal algorithm once for each single order. Since the model is meant to integrate into IKEA's customer-facing website, where customers need to be presented with delivery options for their orders in (near) real-time, this would be too time-consuming. Instead of solving the entire ILP in Subsecton **??**, we solve a slightly simpler version, where we relax all variables that do not directly correspond to the current order to be continuous.

We expect that this will not affect the quality of the solution too much, as solutions will likely tend towards integer values for their variables due to the nature of the formulation. Moreover, any rounding errors are unlikely to significantly influence solution quality, as we are planning around inherently simple and imprecise predictions. The resulting ILP only has seven binary variables and can be solved within a fraction of a second, making it more suitable for potential integration into IKEA's customer-facing website.

We explored the sensitivity of this method to the quality of the prediction of future orders. More specifically, we tested whether a more accurate estimate of the total orders per day would improve the solution. For each day, we used the actual number of orders received that day rather than taking a fixed estimate of 600 orders as a prediction. Aside from this, we keep the method the same, sampling at random from orders in the last fifteen days of the orders dataset.

3.3.2 Shadow Prices

Implicitly, the (I)LP model essentially uses a greedy strategy augmented with shadow prices for depot capacity derived from the ILP for the predicted orders. So, rather than solving the complete ILP for each customer order, one can instead use pre-computed shadow prices and a greedy assignment heuristic. This is more computationally efficient and could improve runtime significantly compared to the basic (I)LP model.

Moreover, using shadow prices may facilitate extensions of our method to more complex settings, e.g., involving order splitting. It may even be possible to use *column generation* to solve even more complicated versions, while still using shadow prices to enable the website to quickly allocate individual orders. The shadow prices could be recomputed throughout the day, taking into account new information.

Our heuristic approach attempts to balance the allocation of orders by defining residual capacities R_u for each store, and adding a penalty to the greedy approach depending on the residual capacity. In a sense, the 'correct' value for the penalty factor that the heuristic approach tries to approximate (by intuition) should be one equal to the shadow prices from the ILP.

3.4 Deep Learning

Our last approach explores the application of deep learning to predict optimal store allocations for incoming orders at IKEA. This method differs from the heuristic algorithms, as it involves offline training and does not necessarily require real-world data. Instead, the neural networks can be trained using optimal solutions of synthetic data. However, to ensure consistency with the other approaches involved in this study, we used the same dataset instead of augmenting the training process with synthetic data.

In this approach, each order is labeled with its optimal solution. This allows the network to learn from the data, capturing patterns that lead to the corresponding decisions. In the end, our goal is to enable the network to generalize this learning process to new orders and predict the best allocation without prior knowledge of all orders for the day.

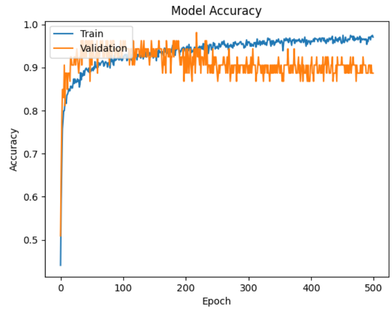
The architecture of our neural network is summarised in Table 1. We aimed to keep the initial model simple, exploring the feasibility of this approach to the problem setting. This allowed us to get initial predictions without the need for more complex architectures.

Component	**Details**
Input Layer	Processes order details and historical data.
Hidden Layers	First: 64 neurons, ReLU activation. Second: 32 neurons, ReLU activation.
Output Layer	Softmax activation.
Loss Function and Optimiser	Categorical Cross-Entropy, Adam optimiser.
Training Parameters	500 epochs, batch size of 10, 10% validation split.

Table 1: Summary of the neural network architecture.

The results of the training process are visualized in Figure 12, displaying the model's accuracy and loss over epochs. The plot shows that the model initially learns quite fast, obtaining high accuracy scores. The training loss shows a steady decrease; however, after around 90 epochs, the validation loss increases, which indicates overfitting. It is, therefore, important to stop training at this point to allow the model to generalize on new unseen orders.

As mentioned before, we recognize that the use of real historical data from IKEA in training our deep learning model was primarily to maintain consistency across all approaches evaluated in this study. The scope of historical data is limited to past scenarios, and while it provides a valuable learning foundation, it may not capture insights into the dynamics of future order fulfillment. Therefore, incorporating syn-



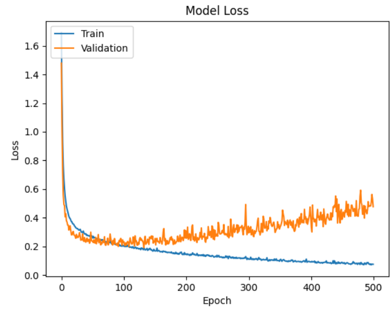


Figure 12: Training and validation accuracy (**left**) and loss (**right**) for the deep learning model over 500 epochs.

thetic data could significantly augment the training process. Such synthetic datasets would not only serve to validate the model's performance against known outcomes but also prepare it for unseen situations, potentially improving its robustness and generalizability.

4 Evaluation and Discussion

In this section, we provide a comprehensive overview of the performance and implications of all implemented order allocation strategies. Additionally, we discuss key points and contextualize our findings.

4.1 Overall Performance

In our study, we evaluated all order allocation strategies by calculating the total cost, the percentage of unfulfilled orders, and their running times using a test dataset. We employed a unified testing framework to run these strategies in a modular manner, ensuring maximal consistency in the results. The data for the study, provided by IKEA, was divided into two sets: a training set (the last 15 days of the month) and a test set (the first 15 days of the month). We computed the evaluation metrics on the test dataset. Table 2 provides a comparison overview of the order allocation strategies.

All execution times were recorded on a shared GitHub Actions runner instance. This instance is a 4-core machine running Linux (Ubuntu 22.04) with 16GB of RAM.

The total cost of each strategy, along with the percentage of its failed orders, is also visualized in Figure 13. Notably, the *deep* strategy (Neural Network) results in

Order Allocation Strategy	Total cost	Unfulfilled orders (%)	Running time (s)
Optimal solution	1,024,164	1.32 %	228.06 s
Greedy algorithm	1,764,998	3.82 %	0.14 s
Heuristic	1,489,415	2.75 %	0.13 s
LP with Prediction	1,242,627	2.07 %	1586.55 s
Neural Network	2,090,043	4.67 %	1.55 s
Greedy with Order Splitting	1,429,178	2.85 %	76.27 s

Table 2: Comparison of all implemented order allocation strategies.

the highest costs and the largest percentage of failed orders, with *greedy* coming in next. *Reapeatedilp* (LP with Prediction) falls on the lower end of the spectrum with cost and failed order percentage compared to the *optimal* strategy. The *heuristic* approach improves over *greedy* while maintaining low execution time. The *splitgreedy* (Greedy with order splitting) strategy incurs a lower total cost and failed percentage of orders than the *greedy* approach. However, a drawback of the order splitting extension compared to regular greedy is its significantly higher running time.

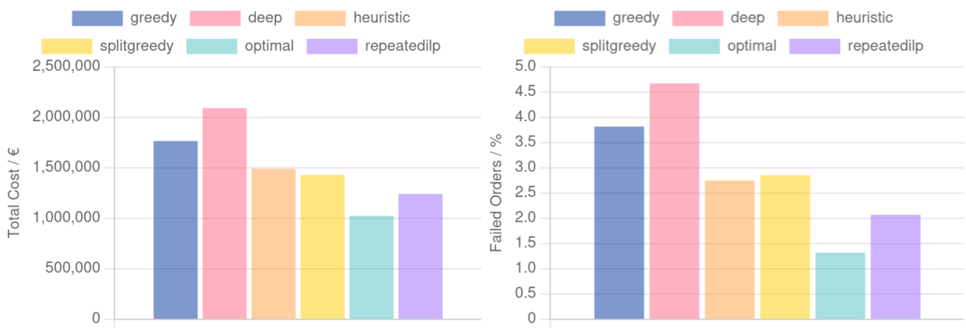


Figure 13: Overall cost and percentage of failed orders for each allocation strategy, as evaluated on the sample dataset provided.

4.1.1 Performance per Day

Figure 14 illustrates the performance for each day of the simulation. The performance varies significantly due to the different sets of orders each day. Notably, the *deep* strategy shows a spike in cost and failed orders on the third day, which was a busy day. This highlights its sensitivity to complex variations in daily orders, particularly on active days. As expected, the *optimal* strategy yields the lowest costs and the fewest failed orders. However, it exhibits significant day-to-day variations in runtime. Similarly, the *repeatedilp* strategy consistently shows prolonged runtimes across all days. This emphasizes the fact that the *repeatedilp* strategy is more computationally expensive compared to the others.

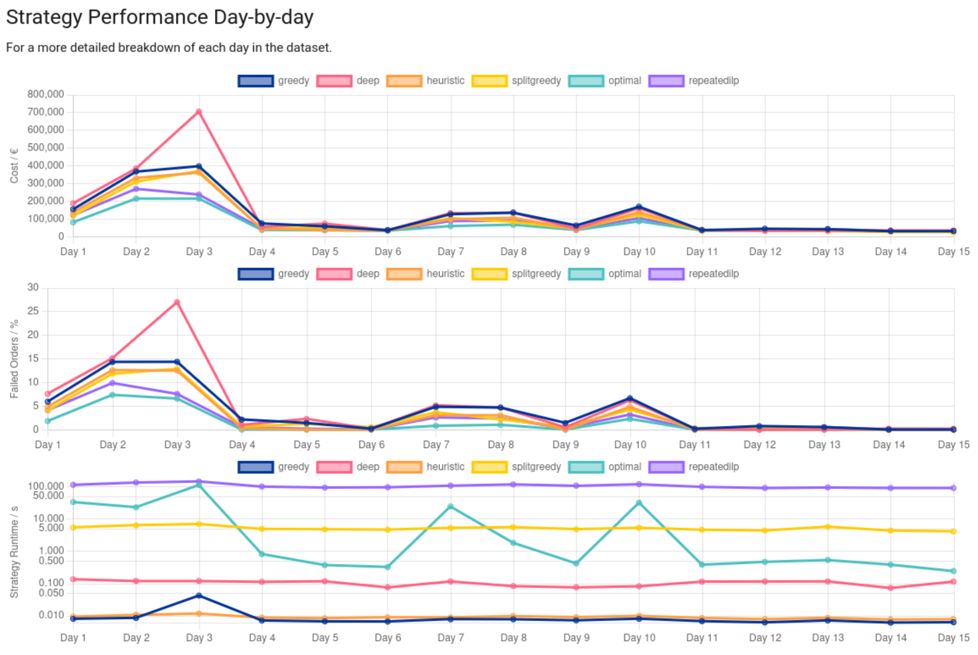


Figure 14: Cost, failed orders, and runtime per day comparison of the various allocation strategies, as evaluated on the sample dataset provided.

4.1.2 Best Solution: (I)LP with Predictions

For the days in the test dataset, IKEA's greedy algorithm incurs a cost of 1,764,998, while the optimal solution has a lower cost of 1,024,164. The ILP, with Prediction methods, has a cost of 1,242,627, so it realizes 522,371, or 76%, of the potential savings.

The experiment using the real number of orders as a prediction resulted in a slightly cheaper solution (cost 1,184,156 versus 1,242,627 for using a fixed estimate of 600 orders per day); see Figure 15. However, the difference compared to using the fixed estimate is very small, so it appears that the method is not sensitive to the accuracy of the number of estimated orders.

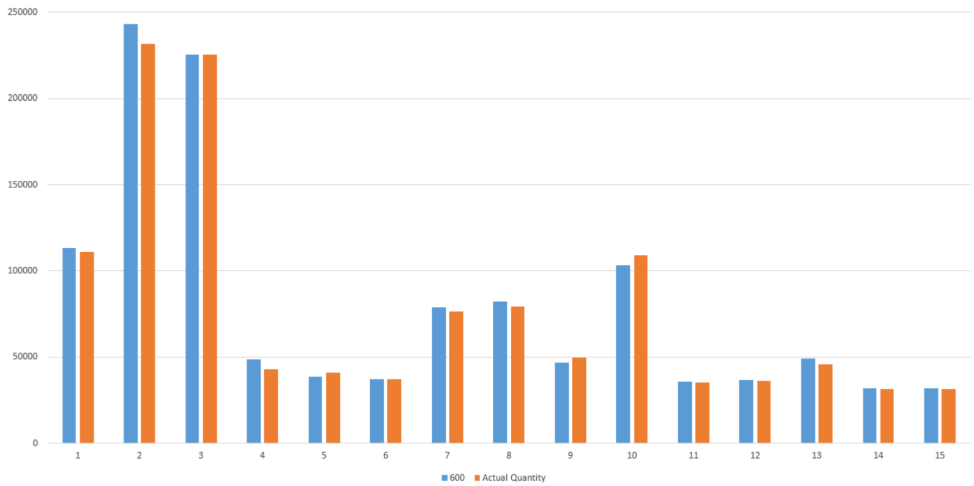


Figure 15: Total cost per day for the (I)LP model under the fixed estimate of 600 orders and the actual quantity of orders taken as an estimate, tested on the first fifteen days of the orders dataset. Taking the actual quantity reduced the total cost by ±2 percent on average.

On most days using the better prediction resulted in lower costs, but on some days we obtained higher costs than the fixed estimate. This included some relatively quiet days – i.e., replacing a (for those days) very inaccurate prediction with a better one resulted in worse performance. It may be the case that the sampled orders during the run with the fixed estimate were coincidentally more similar to the actual orders of that day than during the run with the actual order quantity.

4.1.3 Dashboard

The effectiveness of the various order allocation strategies is dynamically illustrated in our interactive dashboard (Figure 16).

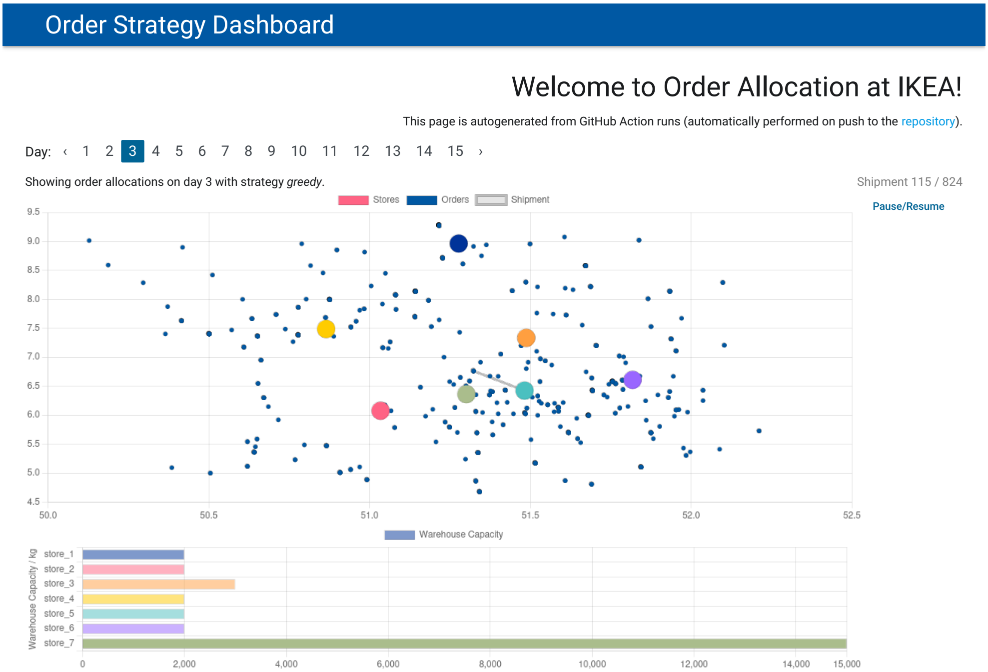


Figure 16: The dashboard created as part of the project. The current screen visualises the order locations (small dots in blue), as well as the store locations (larger dots, coloured). Shipments are indicated by a line connecting an order to a store. This is an interactive visualization as the user can follow the allocations live over time, providing intuition for the individual strategies. The store depletion is shown at the bottom and also tracked over time.

4.2 Batch planning

Between the fully online setting, where each order is planned immediately without information on future orders, and the offline setting, which has information on all orders for the day, there exists an intermediate approach that involves planning a subset of consecutive orders. This method requires the algorithm to wait for a certain number of orders to accumulate before planning them as a batch. However, we did not explore this modeling option as it would not provide a viable solution for IKEA. IKEA's expected service level requires that customers receive delivery options instantaneously, making this approach impractical.

4.3 Algorithm Performance on Quiet Days

From Figure 14, we identify days 4 and 14 as relatively quiet days with lower order volumes. On these days, the performance differences between the various allocation strategies are significantly diminished, and most approaches closely align with the optimal solution.

On day 4, the *greedy* and *splitgreedy* algorithms' cost and percentage of failed orders do not deviate too much from the optimal solution. This is because the lower order volume allowed the *greedy* approach to allocate most orders to their respective optimal locations without exhausting the resources of the closer, more cost-effective stores. The *heuristic* approach and the *repeatedilp* also show minimal deviations from the optimal solution on this quiet day.

It is worth noting that the runtime of the *greedy* approach remains more or less consistent across most days, with an outlier observed on day 3, likely an anomaly and would not persist if the experiments were repeated. On the other hand, the *optimal* runtime fluctuates from day to day, where it peaks on day 3 and then decreases on day 4.

The running time of the other algorithms remains relatively consistent across all days, and even the very quiet Day 14 has no impact on it. In fact, the performance of all the algorithms on day 14 is similar to that observed on days 11-15. This phenomenon is also observed for both cost and percentage of unfulfilled orders, where by day 11, all strategies converge very close to the optimal solution. The simplicity of the order patterns on such quiet days (onwards of day 11) resulted in most strategies giving allocations that were nearly optimal.

These findings suggest that investing in high-complexity strategies (such as *deep*) may not always be justifiable, particularly for IKEA, which experiences fluctuations in daily order volumes. For operations where quiet days are frequent, focusing on fine-tuning simpler algorithms (such as *heuristic*) or developing strategies that can scale complexity according to demand may be a more cost-effective approach.

4.4 Scalability

In evaluating the scalability of our solutions, it's important to consider how they would adapt to changes in scale, such as an increased number of orders, variability in warehouse capacities, or an expansion in the number of warehouses. Scalability is a critical attribute, particularly for a global retailer like IKEA, which must be able to maintain efficiency as it grows and as demand fluctuates.

Our solutions demonstrate scalability potential in terms of computational runtime and adaptability to different problem settings. The *greedy* and *heuristic* strategies, with their lower computational overhead and faster runtimes, seem more scalable for handling a larger number of orders. The *greedy* approach, for instance, focuses on immediate cost minimization, while the *heuristic* incorporates residual capacities, making them versatile across different order volumes and warehouse capacities.

That being said, the scalability of strategies such as *repeatedilp* and *splitgreedy* could be challenged by larger datasets and a more extensive network of warehouses. The computational complexity of these methods, as indicated by longer runtimes, may lead to diminishing returns as order volume increases unless parallelization or other efficiency-enhancing measures are implemented. Additionally, these strategies might require more sophisticated hardware and software infrastructure to handle the increased complexity of larger-scale operations.

Additionally, since our *repeatedilp* method combines a greedy strategy with shadow prices for depot capacity, rather than solving the entire LP for each individual order, we can use precomputed shadow prices along with a greedy assignment heuristic. This approach potentially improves computational efficiency and opens up possibilities for extending the method to handle more complex settings. Other optimization techniques, such as column generation, can be used to tackle highly complicated scenarios while still maintaining the ability to swiftly allocate orders on the website.

The capacity of warehouses is another critical factor for scalability. If IKEA were to increase warehouse capacities, algorithms would need to adapt to these new constraints. Strategies that incorporate predictions or utilize deep learning may need retraining or recalibration to account for the changes in capacity and ensure that order allocations remain efficient. Furthermore, the introduction of more warehouses could complicate the order allocation process due to the increased number of potential allocation points. Strategies that performed well with a smaller number of warehouses might not scale as well with the addition of more locations due to the increase in possible combinations for order distribution.

In our analysis, we assumed a static warehouse network. In reality, IKEA's warehouse network may expand or contract. Our solutions might also need to consider other logistic aspects of the problem such as varying delivery times and time windows of customers in order to maintain high service levels as the scale and complexity of operations increase.

4.5 Deep and Machine Learning Potential

With the current *deep* model trained on only 15 days' worth of data, the findings show poor performance and possibly overfitting as seen from Figure 12. However, its performance could significantly improve with access to a larger, more diverse dataset.

Neural networks thrive on large datasets that span several months or years as they allow the model to capture complex patterns in customer behavior, seasonal variations, and order distributions that are not evident in smaller samples. Training the neural network with data containing various seasonal peaks, marketing campaigns, and a broader range of customer orders would result in a more accurate and robust model. The model could then not just improve the optimal warehouse prediction for order fulfillment but also anticipate fluctuations in order volume, which are vital for capacity planning and inventory management.

It is important to note that while the neural network might inherently capture the behavior of busyness through the trained daily orders, this aspect is not explicitly modeled as an input feature. Incorporating features that directly relate to the characteristics of busy days or other relevant input parameters could potentially enhance the model's robustness and accuracy.

Similarly, for the forecasting of daily orders, our current time series models are constrained by the limited scope of a single month's data. With this timeframe, it is challenging to capture the long-term seasonality and trends that could impact order volumes. Expanding the dataset to include multiple years would allow us to incorporate annual patterns, such as holiday seasons, back-to-school periods, or other cyclical events that significantly influence customer purchasing behaviors. This enriched data would increase the accuracy of the forecasting models. For instance, an improved forecast of daily orders could directly benefit the *repeatedilp* strategy, which currently relies on a rough average of daily orders. By integrating a machine learning approach, the updated strategy could dynamically adjust to expected daily fluctuations, optimizing resource allocation in real-time.

Lastly, more extensive data would enable better feature engineering and hyperparameter optimization, which could lead to more sophisticated models. These could include ensemble methods that combine various predictive techniques to account for different types of variability in order data, such as random forests for feature-rich insights or gradient boosting machines for performance efficiency.

4.6 Practical Considerations

One of the primary considerations is the geographical diversity inherent in a global operation like IKEA's. The dataset used for our analysis is potentially limited in scope, representing a specific regional market or set of logistic circumstances. Other regions may present unique challenges, such as varied delivery infrastructure, different urban density profiles, or diverse customer behavior patterns, all of which could impact the effectiveness of the order allocation algorithms. Therefore, models must be tailored or adapted to reflect the geographical specifics of each new dataset they are applied to.

Running these allocation strategies in real time presents another layer of complexity, particularly concerning computational resources and cost. Strategies like the *repeatedilp* and *splitgreedy* are computationally intensive and would require significant processing power to operate in a real-time environment, potentially leading to increased operational costs. Ensuring that the response time of these algorithms meets the real-time needs of an e-commerce environment may require investment in high-performance computing resources or cloud-based infrastructures.

4.7 Generalisability: Thinking Beyond IKEA

The inclusion of a 'busy day factor' in our heuristic approaches demonstrates adaptability to varying daily demands, which can be an underlying feature in many operational settings. Moreover, the concept of order splitting is an aspect of our problem that has broad applicability, such as in manufacturing where components of a product may need to be sourced from multiple suppliers, and the decision on which suppliers to use could be optimized using a similar approach, especially when considering factors like cost, distance and supplier reliability.

Furthermore, the fundamental assumptions and constraints used in our models, such as residual capacity and single-store constraints, are adaptable to problems where resources are finite and must be distributed in an efficient manner. These constraints can be further modified to fit the specifics of other operational challenges, whether it is in inventory management, distribution of services, or even project task allocations within large organizations.

Therefore, in essence, our approaches do not only address the immediate optimization needs of IKEA's e-commerce order fulfillment process but also present a malleable framework for optimal and heuristic approaches to a fundamental class of assignment problems found across many operational sectors.

5 Conclusions

In this study, we investigated various strategies for optimizing the allocation of online orders to fulfillment centers in the context of IKEA's e-commerce operations. Our goal was to minimize total costs while maximizing the number of fulfilled orders. We formulated the problem as an integer linear program and developed several solution approaches, including a greedy algorithm, a heuristic based on residual capacities, an LP with predictions, and a deep learning model.

Through extensive computational experiments on a real-world dataset provided by IKEA, we evaluated the performance of these strategies in terms of total cost, percentage of unfulfilled orders, and running time. The results demonstrated that the *repeatedilp* approach achieved the best overall performance, realizing 76% of the potential savings compared to IKEA's current *greedy* allocation strategy. The *heuristic* method also showed promising results, offering significant improvements over the *greedy* approach while maintaining low computational overhead.

Returning to our initial research questions, we can now provide the following insights:

RQ1 The different allocation strategies had a significant impact on both the feasibility and costs of order fulfillment. The *repeatedilp* and *heuristic* approaches outperformed *greedy* in terms of total cost and percentage of fulfilled orders.

RQ2 The optimal offline solution, which assumes perfect knowledge of all orders in advance, provided a benchmark for evaluating the potential for optimization. Our results showed that there is considerable room for improvement compared to the current greedy approach, with the *repeatedilp* realizing a substantial portion of the potential savings.

RQ3 We did not explicitly compare batch processing to immediate allocation strategies due to the real-time requirements of IKEA's online platform.

RQ4 The *splitgreedy* approach demonstrated improved performance compared to the standard greedy approach in terms of cost and percentage of failed orders. However, the increased computational complexity and longer running time of this strategy highlight the trade-off between order fulfillment reliability and operational efficiency.

RQ5 Our analysis revealed that factors such as the daily order volume and the capacity of fulfillment centers play a crucial role in the performance of the allocation strategies. On quieter days, the performance gap between the different methods diminishes, indicating that the potential benefits of more sophisticated strategies may be limited in such scenarios.

We also observed that the performance gap between the allocation strategies diminishes on quieter days with lower order volumes. The *deep* model, despite its potential, was found to be sensitive to the limited training data available. We also discussed the scalability of the proposed methods, highlighting the importance of considering factors such as geographical differences, real-time computational requirements,

Furthermore, we emphasized the generalisability of our problem formulation and solution approaches to various other domains beyond retail. The core concepts and techniques presented in this work can be adapted to any setting that involves the assignment of orders or jobs to capacitated resources, making our contributions relevant to a broad range of optimization problems.

Future research directions include expanding the dataset to cover a longer time period and more diverse scenarios, which would enable the development of more sophisticated forecasting models and improve the performance of the *deep* approach. Incorporating additional real-world constraints, such as varying delivery times and customer preferences, could enhance the practicality of the proposed methods. Investigating the potential of integrating our order allocation optimization with other operational decisions, such as inventory management and transportation planning, could lead to a more holistic and efficient supply chain management strategy.

In conclusion, this study provides valuable insights for practitioners seeking to improve their online order fulfillment processes. By leveraging optimization techniques and machine learning, retailers can significantly reduce costs, increase customer satisfaction, and gain a competitive edge in the rapidly evolving landscape of online commerce. Our work lays the foundation for further research and innovation in this exciting field.

Authorship Statement. Author TZ contributed primarily to the '(I)LP with Predictions' method (Sections 3.3 and 4.1.2) and takes responsibility for only this part of the scientific product.

Acknowledgements. We thank Jan de Munck and Arturo Perez Rivera for providing an interesting case and the SWI organizers for the pleasant atmosphere during the workshop.

Conflicts of Interest. The authors declare no conflicts of interest.

References

Alaei, S., Hajiaghayi, M., and Liaghat, V. (2013). The online stochastic generalized assignment problem. In *International workshop on approximation algorithms for combinatorial optimization*, pages 11–25. Springer. https://doi.org/10.1007/978-3-642-40328-6.

Albareda-Sambola, M., Van Der Vlerk, M. H., and Fernández, E. (2006). Exact solutions to a class of stochastic generalized assignment problems. *European Jour-*

nal of Operational Research, 173(2):465–487. https://doi.org/10.1016/j.ejor.2005.01.035.

Beliën, J., Boute, R., Creemers, S., De Bruecker, P., Gijsbrechts, J., Padilla Tinoco, S. V., and Verheyen, W. (2017). Collaborative shipping: Logistics in the sharing economy. *ORMS Today*, 44(2):20–23. https://pubsonline.informs.org/do/10.1287/orms.2017.02.01/full.

Berger, V. W. and Zhou, Y. (2014). Kolmogorov - smirnov test: Overview. https://dx.doi.org/10.1002/9781118445112.stat06558.

Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer programming*. Springer. https://doi.org/10.1007/978-3-319-11008-0.

Croxton, K. L. (2003). The order fulfillment process. *The International Journal of Logistics Management*, 14(1):19–32. https://doi.org/10.1108/09574090310806512.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press. https://dl.acm.org/doi/10.5555/3001460.3001507.

European Environmental Agency (2024). Transport and mobility. Accessed: 1 Feburary 2024 https://www.eea.europa.eu/en/topics/in-depth/transport-and-mobility.

Garza, A. and Mergenthaler-Canseco, M. (2023). TimeGPT-1. https://arxiv.org/abs/2310.03589.

Ishfaq, R. and Bajwa, N. (2019). Profitability of online order fulfillment in multi-channel retailing. *European Journal of Operational Research*, 272(3):1028–1040. https://doi.org/10.1016/j.ejor.2018.07.047.

Jin, X. and Han, J. (2010). *K-Means Clustering*, pages 563–564. Springer US, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_425.

Levin, M. (2023). A real-time control policy to achieve maximum throughput of an online order fulfillment network. *Transportation Science*. https://doi.org/10.1287/trsc.2023.0096.

Li, Z., Wang, H., and Yan, Z. (2023). Sample-based online generalized assignment problem with unknown poisson arrivals. https://arxiv.org/abs/2302.08234.

Morton, D. P., Bard, J. F., and Wang, Y. M. (2009). Solving a stochastic generalized assignment problem with branch and price. *Computational Biology: New Research*, pages 99–128. https://pitt.primo.exlibrisgroup.com/permalink/01PITT_INST/e8h8hp/alma9960970133406236.

Onal, S., Zhu, W., and Das, S. (2023). Order picking heuristics for online order fulfillment warehouses with explosive storage. *International Journal of Production Economics*, 256:108747. https://doi.org/10.1016/j.ijpe.2022.108747.

Öncan, T. (2007). A survey of the generalized assignment problem and its applications. *INFOR: Information Systems and Operational Research*, 45(3):123–141.

Rao, S., Griffis, S. E., and Goldsby, T. J. (2011). Failure to deliver? linking online order fulfillment glitches with future purchase behavior. *Journal of Operations Management*, 29(7-8):692–703. https://doi.org/10.1016/j.jom.2011.04.001.

Ricker, F. and Kalakota, R. (1999). Order fulfillment: the hidden key to e-commerce success. *Supply Chain Management Review*, 11(3):60–70. https://api.semanticscholar.org/CorpusID:51256454.

Sarin, S. C., Sherali, H. D., and Kim, S. K. (2014). A branch-and-price approach for the stochastic generalized assignment problem. *Naval Research Logistics (NRL)*, 61(2):131–143. https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.21571.

Spoerl, D. and Wood, R. K. (2003). A stochastic generalized assignment problem. In *INFORMS Annual Meeting, Atlanta, GA*, volume 19, page 22. https://chds.academia.edu/DavidSpoerl.

Tarsitano, A. and Amerise, I. L. (2017). Short-term load forecasting using a two-stage sarimax model. *Energy*, 133:108–114. https://doi.org/10.1016/j.energy.2017.05.126.

United Nations (2024). Take urgent action to combat climate change and its impacts. Accessed: 1 Feburary 2024 https://sdgs.un.org/goals/goal13#progress_and_info.

Wang, Y., Wang, T., Wang, X., Deng, Y., and Cao, L. (2023). Data-driven order fulfillment consolidation for online grocery retailing. *INFORMS Journal on Applied Analytics*. https://doi.org/10.1287/inte.2022.0068.

Xie, Y., Ma, W., and Xin, L. (2023). The benefits of delay to online decision-making. http://dx.doi.org/10.2139/ssrn.4248326.

Zhao, Y., Wang, X., and Xin, L. (2022). Multi-item online order fulfillment in a two-layer network. *Chicago Booth Research Paper*, (20-41). http://dx.doi.org/10.2139/ssrn.3675117.