Tricks from the Trade for Large-Scale Markdown Pricing: Heuristic Cut Generation for Lagrangian Decomposition

Robert Streeck*[1], Torsten Gellert[1], Andreas Schmitt[1], Asya Dipkaya[1], Vladimir Fux[1], Tim Januschowski[1], and Timo Berthold[2]

[1]Zalando SE, Germany
[2]TU Berlin, Germany

Abstract

In automated decision making processes in the online fashion industry, the 'predict-then-optimize' paradigm is frequently applied, particularly for markdown pricing strategies. This typically involves a mixed-integer optimization step, which is crucial for maximizing profit and merchandise volume. In practice, the size and complexity of the optimization problem is prohibitive for using off-the-shelf solvers for mixed integer programs and specifically tailored approaches are a necessity. Our paper introduces specific heuristics designed to work alongside decomposition methods, leading to almost-optimal solutions. These heuristics, which include both primal heuristic methods and a cutting plane generation technique within a Lagrangian decomposition framework, are the core focus of the present paper. We provide empirical evidence for their effectiveness, drawing on real-world applications at Zalando SE, one of Europe's leading online fashion retailers, highlighting the practical value of our work. The contributions of this paper are deeply ingrained into Zalando's production environment to its large-scale catalog ranging in the millions of products and improving weekly profits by millions of Euros.

1 Introduction

Online retailers face challenges that are especially important in periods where economies stagnate and companies focus on efficiency: they need to guarantee a smooth experience throughout the whole customer journey, organize distribution of large assortments (in the millions of products) and efficiently manage inventory across multiple countries. A crucial aspect for the success of an online retailer is its capability to dynamically scale commercial processes to account for sales periods, traffic growth and external market conditions. This puts strict constraints on the underlying algorithms and systems, which need to deliver results in tight timelines (minutes or hours), even if suboptimal, to ensure business continuity. The large assortment necessitates a high degree of automation so that human oversight cannot be guaranteed for both scale and response time requirements. Key operational questions are hence fully automated. A prime example for such a system which is both operationally and commercially critical and is subject to scaling and response-time requirements is markdown pricing.

The goal in markdown pricing is to maximize profit while managing the fixed inventory for the entirety of a season.

*robert.streeck@zalando.de

In this paper, we consider large-scale price optimization systems for online retailers and the challenges that arise. Our show case throughout the paper will be Zalando SE, a leading online fashion retailer in Europe, throughout. With more than 50 million customers, presence in over 25 European countries and close to 2 million articles in its assortment, efficient pricing is one of the cornerstones of the Zalando business. To be successful, pricing systems and algorithms need to be

1. scalable, as the assortment is constantly growing to account for varying tastes and demands for novel propositions
2. fast, as the price update cycle has to meet commercial needs and match competitors' pacing.

In addition, markdown pricing is one of the most effective levers to boost various dimensions of commercial performance (discounts depth, revenue, profit, etc.), both in in specific countries/-markets and on a company-wide level. This requires the pricing system to be able to steer towards strategic company goals, which in turn necessitates an integrated optimization (due to linking constraints) for all, or at least larger sets of markets. This poses an additional challenge for algorithm design, since the aforementioned commercial needs demand timely execution.

A typical approach in markdown pricing follows the predict-then-optimize paradigm (see e.g., (Ferreira et al., 2016; Loh et al., 2022; Kedia et al., 2020; Caro and Gallien, 2012) for examples in pricing) where a number of forecasting models (e.g., (Kunz et al., 2023; Schultz et al., 2023) for forecasting for pricing at Zalando and (Salinas et al., 2020; Lim et al., 2021; Rasul et al., 2021) for more general examples) provide input to an optimization problem. In practice, mixed integer programming (MIP) solutions are often employed given their adaptability to changing constraints (Kedia et al., 2020; Li et al., 2022; Caro and Gallien, 2012). The pricing approach of our show case Zalando follows this paradigm and here, we will consider in particular the "optimize" part. Previous work (Li et al., 2022) described the basic setup of the pricing optimization problem at Zalando and key ingredients necessary for its solution, such as a Lagrangian decomposition framework, modelling approaches and basic heuristics. However, in the years since the deployment of this solution in production, roughly since 2021, there were occasional repercussions from unfavourable practical situations that led to non-convergence of the solution. As a further motivation for improving the heuristic performance of the markdown-pricing system, the scale of large online retailers like Zalando implies that even modest percentual gains in finding more profitable prices result in large absolute monetary gains – easily in the multi-digit million Euro range.

The main contribution of this paper are extensions of the pricing algorithm described by Li et al. in (Li et al., 2022) via novel combinations of existing heuristic techniques both for the Cutting Plane iterations and primal solution construction. The focus is on providing high quality, but potentially sub-optimal solutions within tight time boundaries as commonly dictated by commercial processes. These extensions address the aforementioned challenges around scalability and speed. We also show in empirical evaluations that our approach allows to improve revenue by 3%–6% and profit by 2%–5% for our show case Zalando SE while having negligible impact on solution speed.

Our paper is structured as follows. In Section 2, we discuss work related to this paper. In Section 3, we introduce the necessary notation and give the formalization of the markdown pricing problem. Section 4 discusses the algorithms used to solve the markdown pricing problem and in Section 5, we present the main methodological novelties of this paper, namely the heuristics for solving the markdown problem. Section 6 provides extensive experimental results. We conclude in Section 7.

2 Related Work

Pricing and revenue management are mature areas of research stretching multiple disciplines, see e.g., (Phillips, 2021) for an introduction to the topic from an Economics perspective. The Machine Learning community typically covers aspects involving prediction, like demand forecasting, e.g., (Eisenach et al., 2020; Kunz et al., 2023; Oreshkin et al., 2020) and sometimes also taking a reinforcement learning perspective (e.g., (Madeka et al., 2022) for an example of an industrial application). Econometricians tend to focus on causal aspects or price elasticity estimation (Athey and Imbens, 2007; Strauss et al., 2018; de Chaisemartin and D'Haultfœuille, 2020; Deaton and Muellbauer, 1980; Fogarty, 2010; Hughes et al., 2008; DeFusco and Paciorek, 2017) and in the formulation of the optimization problems needed for profit-optimal pricing (Phillips, 2021). Here, we take an Operations Research erspective (see (Ferreira et al., 2016; Li et al., 2022) for examples), where we take both the predictive problems and the problem formulation as given and rather work on solving the optimization problem most efficiently within the requirements dictated by production purposes. This is, to the best of our knowledge, under-explored in this specific form in pricing. Chen et al. (2024) present a notable exception with a recent contribution, however, not for markdown pricing but for multi-product pricing in a ride-hauling setting.

Beyond pricing, in the larger context of mathematical optimization, heuristic methods are omnipresent, see (Berthold, 2014a) for an overview. For complex MIP problems, a plethora of general-purpose heuristics exist, see, e.g., (Berthold, 2014b; Berthold et al., 2019; Fischetti and Lodi, 2003; Gamrath et al., 2019). Also, cut generation, aggregation, and selection strategies are often of a heuristic nature, consider, e.g., (Andersen et al., 2005; Bonami et al., 2008; Caprara and Fischetti, 1996; Turner et al., 2023) and in particular, (Fischetti and Salvagnin, 2011) which heuristically generates cutting planes from a Lagrangian relaxation of a general MIP. While our approach is general in the sense that it can be easily adapted to other applications of using Lagrangian relaxations to solve MIPs, we believe that the particular tuning of thresholds, limits, and selection criteria would not generalize as easily.

3 Background and Problem Formulation

The pricing problem we tackle can be formalized as follows. For an assortment of n articles, we want to determine for each article $i \in N = \{1, \dots, n\}$ a price x_i to maximize the long-term profit (LTP) $\max_{x_i \in \mathcal{X}_i} f_i(x_i)$. Here, the set $\mathcal{X}_i$, with liberty in the notation, denotes the feasible set of a mixed integer programming formulation for the article prices, described by (linear) constraints, bounds, and auxiliary variables not explicitly mentioned. Similarly, the linear objective function $f(x)$ depends on auxiliary variables. The notation is deliberately kept general to demonstrate the broad possible applications. For the showcase of Zalando, this entails that x_i models varying prices for the article in different countries and various (future) weeks. Auxiliary variables model, amongst others, the sales and the stock of the article over the future weeks. In this showcase, we consider around 500 000 articles. For a single article, the model contains up to 10 000 variables (5 000 binary) and 10 000 constraints. More details on the formulation of the pricing problem at Zalando can be found in (Li et al., 2022).

Additional business constraints restrict the solution space. Typical examples are targets for an average discount rate or a gross merchandising volume, both aggregated over each country. These targets are often refereed to as *linking constraints*, as they depend on all articles. In a general form, the primal pricing problem is thus given as

$$P = \max_{\substack{x \in \mathcal{X} \\ Ax \leq b}} \sum_{i \in N} f_i(x_i), \tag{1}$$

where $Ax \leq b$ enforces the linking constraints and $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ describes the Cartesian product of the individual articles feasible sets.

We again refer to (Li et al., 2022) for a comprehensive descriptions of the linking constraints and the basic Lagrangian algorithm which we summarize in the next section. For a concrete example of linking constraints, consider the average discount rate over all articles per given country which we typically constrain to a certain range as a leverage for business leaders to steer prices. The *sales weighted discount rate* extends the discount rate by weighing discounts by sales,

$$\text{sDR} = \frac{\sum_i (p_i - \bar{x}_i) \cdot s_{i,\bar{x}_i}}{\sum_i p_i \cdot s_{i,\bar{x}_i}}$$

where p_i denotes the un-discounted price of article i and $s_{i,\bar{x}_i}$ the number of items sold of article i when offered for price $\bar{x}_i$, and the set of variables $\bar{x}_i$ is a subset of the decision variables x_i.

Lagrangian Relaxation

With typical sizes of online retailer assortments ranging in the millions of articles (which would result in billions of variables in our problem), the pricing problem is prohibitively large for off-the-shelf solvers. Techniques such as decomposition or aggregation become inevitable for identifying high-quality solutions in practice.

A standard technique for decomposition approaches is based on a Lagrangian relaxation. Here, the violation of each linking constraint is penalized using Lagrangian multipliers $\lambda \geq 0$ in the objective

$$\text{LR}(\lambda, x) = \sum_{i \in N} f_i(x_i) - \lambda^\top (b - Ax).$$

For this modified objective the Lagrangian relaxation searches for an optimal solution x for given λ, leading to the optimization problem

$$\text{LR}(\lambda) = \max_{x \in \mathcal{X}} \text{LR}(\lambda, x) = \sum_{i \in N} \underbrace{\max_{x_i \in \mathcal{X}_i} (f_i(x_i) + \lambda^\top A_i x_i)}_{\text{single article problem}} - \underbrace{\lambda^\top b}_{\text{constant offset}}$$

By relaxing the linking constraints, we end up with one independent optimization problem per article as presented at the beginning of this section. An interesting aspect in solving the optimization problem comes via business requirements. For example, throughout a week, Zalando needs to re-optimize prices several times for its entire assortment of several hundred thousands of articles. The window for taking the markdown pricing decision is determined by the time when the most recent data becomes available and commercial steering meetings. This is typically a few hours. In practice, a single article's optimal long-term profit can be computed well below 10 seconds on a single machine. Via parallelization, the entire assortment can be solved in acceptable time (not exceeding a few hours), when enough computational resources are provided, i.e., a cluster of machines which provides around 1000 cores, provisioned by a cloud provider. The memory required for optimization of the entire assortment easily extends several hundreds of gigabyte.

4 Overview of Algorithms

The objective value of the Lagrangian relaxation yields an upper bound on the pricing problem for every $\lambda \geq 0$. The optimization challenge is to find a combination of multipliers that provides a smallest upper bound, via the Lagrangian multiplier problem $\mu^* = \min_{\lambda \geq 0} \text{LR}(\lambda)$. We will describe how we managed to find suitable multipliers for the markdown pricing problem in the following.

Cutting Plane Procedure

There are different approaches to obtain μ^*, e.g., Subgradient or Bundle methods, see Guignard (2003) for an overview. Here, we follow the approach by (Li et al., 2021) and use a Cutting Plane procedure. To motivate this approach, note that the multiplier problem to compute μ^* can be reformulated as a linear problem where each constraint corresponds to a feasible solution $x \in \mathcal{X}$ of the pricing problem as

$$\mu^* = \min_{\lambda \geq 0, \mu} \mu \text{ s.t. } \mu - (b - Ax)^\top \lambda \geq \sum_{i \in N} f_i(x_i) \ \forall x \in \mathcal{X}. \tag{2}$$

Since every point in $\mathcal{X}$ is a mixed-integer solution to a linear system of inequalities, it is computationally hard to obtain all constraints and therefore the model explicitly: the number of solutions can be exponential in the number of variables.

The Cutting Plane approach avoids this problem by considering a relaxation to Problem (2), which contains only a subset of these potentially exponentially many constraints. The relaxation is tightened in every iteration by adding further valid inequalities. In more detail, assume that in iteration j a number of inequalities determined by points $X^1, \ldots, X^j \in \mathcal{X}$ is included in the relaxation. Then the relaxation is given by

$$\mu^j, \lambda^j = \underset{\lambda \geq 0, \mu}{\arg\min}\, \mu \text{ s.t. } \mu \geq \sum_{i \in N} f_i(X_i^k) + \lambda^\top (b - AX^k) \ \forall k \leq j. \tag{3}$$

The solution to (3) yields a lower bound on μ^* and is simple to solve, see below. To check whether the relaxation solution solves the original problem, i.e., whether μ^j, λ^j belong to the feasible set of Problem (2) it suffices to search for a point $x \in \mathcal{X}$ such that the respective inequality

$$\mu - (b - Ax)^\top \lambda \geq \sum_{i \in N} f_i(x_i) \tag{4}$$

is violated for μ^j, λ^j. The separation problem is equivalent to computing $\mathrm{LR}(\lambda^j)$ and comparing its value against μ^j. If $\mathrm{LR}(\lambda^j) > \mu^j$, the maximizing solution of $\mathrm{LR}(\lambda^j)$ is added as X^{j+1}. In the next iteration this leads to another inequality in the relaxation cutting off the current solution μ^j, λ^j. Otherwise the optimal Lagrangian solution value is found.

This approach enables us to evaluate the quality of our relaxation in each iteration. On the one hand, the value $\min_{k \leq j} \mathrm{LR}(\lambda^k)$ yields an upper bound on μ^* and also P and is therefore called the *dual bound.* On the other hand, the *relaxed primal bound* μ^j yields a lower bound to μ^*, as a relaxation is solved.

It is worth mentioning that the relaxation of the Cutting Plane problem (3) itself is very easy to solve. It is a linear problem with one variable for each linking constraint and with j constraints. In contrast, computing violated cuts involves higher computational effort, since a single new cut requires all subproblems of the entire assortment to be solved. Therefore, it's crucial to aim for a small number of iterations to reach convergence.

Practical considerations for the Cutting Plane procedure

To avoid that the relaxations given by (3) are unbounded during the first iterations, an upper bound $\lambda \leq \bar{\lambda}$ is added to the linear problem. This upper bound is also useful in case a linking constraint is infeasible. The corresponding multiplier would otherwise grow indefinitely. We choose the upper bound $\bar{\lambda}$ large enough that nearly all feasible linking constraints should have a non-violating solution to the Lagrangian relaxation with $\bar{\lambda}$. In consequence, we start the algorithm with $\lambda^0 = 0$, and it will choose $\lambda^1_\ell = \bar{\lambda}$ for all linking constraints ℓ violated by X^0. In cases with a large number of linking constraints (and consequently a high-dimensional Cutting Plane problem) it might require a large number of iterations and added cuts until all occurrences of $\lambda^j_\ell = \bar{\lambda}$ for at least one ℓ have been dropped from the cutting plane problem. Indeed, in practice we observe that most early iterations will contain multipliers at $\bar{\lambda}$, even though such multipliers over-fulfil on the constraints and are likely too high.

Primal Heuristic

The Cutting Plane procedure yields in each iteration a potential solution X^k with price suggestions for the assortment. However, even if the procedure is converged and the optimal value of the Lagrangian Relaxation is found, in practice the solutions commonly violate the linking constraints. This problem is amplified by the fact that there is not sufficient time to perform all iterations necessary for convergence.

To obtain a good solution from the information collected during the Cutting Plane procedure we use a MIP in order to provide a solution close to the optimum with only a minimal amount of constraint violations.

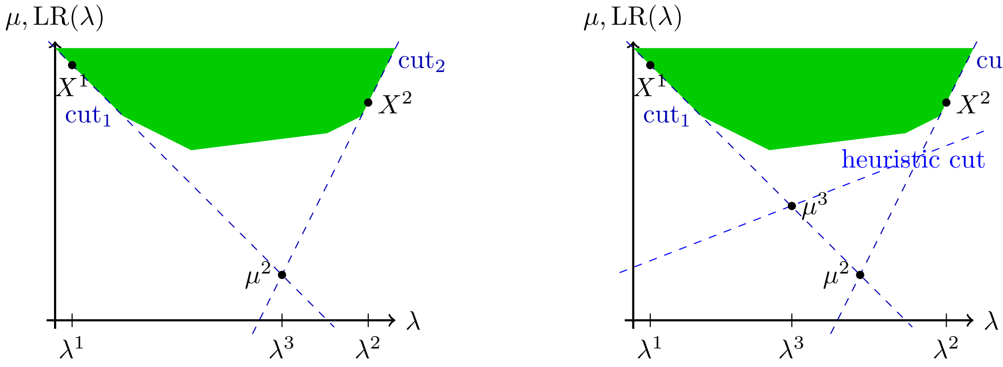
Given the solutions X^k for $1 \leq k \leq j$, the MIP selects for each article i an iteration $k_i \leq j$ such that the offer x_i given by $x_i = X_i^{k_i}$ minimizes a combination of linking constraint violation and profit loss scaled by the highest violation per linking constraints and highest profit observed during the Cutting Plane procedure. The resulting MIP is then

$$\begin{aligned} \max\ & \bar{p} f(x) + \bar{v}^\top \delta \\ \text{s.t. } & \delta \leq Ax - b \\ & x_i = \sum_{1 \leq k \leq j} X_i^k y_{ik} \qquad \forall 1 \leq i \leq n \\ & \sum_{1 \leq k \leq j} y_{ik} = 1 \qquad \forall 1 \leq i \leq n \\ & y_{ik} \in \{0,1\}, \delta \leq 0, \end{aligned} \tag{5}$$

where the binary variables y_{ik} are one if and only if article i takes the values of iteration k and each δ_ℓ measures the violation of the linking constraint ℓ. The constants $\frac{1}{\bar{p}} = \max_k f(X^k)$ and $\frac{1}{\bar{v}_\ell} = \max_k A_\ell^\top X^k - b_\ell - \min_k A_\ell^\top X^k - b_\ell$ for each linking constraint ℓ are furthermore used as scaling factors. This MIP is easier to solve than the original primal problem (1), since the interaction of variables for one article described by $\mathcal{X}_i$ and f_i is already evaluated. We refer to Li et al. (2022) for an in-depth explanation of this final feasibility step.

5 Heuristics for the Markdown Pricing Problem

In this section, we enhance the previously outlined general methodology by introducing a new strategy for identifying violated inequalities for the relaxation. As we will demonstrate, this enhancement not only improves the algorithm's performance but also addresses potential convergence issues. This improvement can have a substantial impact on practical outcomes, leading to a significantly lower objective, which translates into improved long-term profitability.



(a) With two solution for multiplier λ^1 and λ^2 the Cutting Plane procedure chooses multiplier λ^3 for the next iteration.

(b) An additional cut improves the bound μ^2 provided by the cutting plane bound and picks a different next multiplier λ^3.

Figure 1: After evaluating two multipliers λ^1. λ^2, we obtain solutions X^1 and X^2. Their implied inequalities reside on the convex shape, which is implicitly given and not known. The two generated cuts provide a lower bound for μ^* given by μ^2 (left). An additional *heuristic cut* can be created by arbitrary solutions (right). If it cuts off the next multiplier candidate μ^2, we improve the gap without evaluating LR(λ). This cut will most likely not be a facet of the unknown boundary.

The original Cutting Plane procedure obtains cuts by solving LR(λ). In our setting this involves heavy usage of (distributed) computational resources because determining objectives $f_i(x_i)$ still is costly – each models the long term profit for one article for a season and includes the determination of prices and sales in various countries for all weeks of the season. Thus, we investigate heuristic methods to obtain valid cuts with less overhead, i.e., given λ^j and μ^j we would like to find $x \in \mathcal{X}$ such that Inequality (4) is violated for the current iterations multipliers. Figure 1 illustrates this. The generation of such heuristic cuts has to trade-off speed vs. potential much stronger improvements by computing LR(λ^j). Furthermore, changing the strategy has a down-stream impact as the evaluated solutions X^k are used to construct the final solution, see the last part of the previous section.

The core of our idea is to efficiently combine past found solutions $X^1, \dots, X^j$ for which $f_i(X_i^k)$ is already computed for every iteration k and article i . In that case we can find a new solution X^{j+1} which is a combination of past solutions such that for all i there is a $k \in 0, ..., j$ with $X_i^{j+1} = X_i^k$ Importantly, X^{j+1} is a valid solution in $\mathcal{X}$, but not necessarily optimal for LR(λ, x) under any Lagrangian multiplier λ. Thus, there is no guarantee for finding a cut that maximizes the violation or even finding a violated cut at all.

Several strategies can be thought of to obtain X^{j+1}. We investigate the following three:

- Random heuristic: For each article i sample X_i^k uniformly from $k \leq j$. This gives us a baseline to compare the other two strategies.
- Maximum violation heuristic: Restrict the separation problem solved, i.e., the computation of LR(λ^j) to already evaluated price strategies

$$\sum_{i \in N} \max_{x_i \in \{X_i^k : k \leq j\}} f_i(x_i) - \lambda^\top A_i x_i. \tag{6}$$

Since past evaluations of $f_i(X_i^k)$ as well as the contribution of each article to the linking constraints $A_i X_i^k$ are stored, this can be done efficiently by reweighing the violations and sorting.

- Feasibility heuristic: Execute the primal heuristic from the previous section, using the current set of solutions (obtained from the preceding iterations). This process strives to derive a cut from a feasible and almost optimal solution. This approach is the most computationally involved among the three discussed strategies. Moreover, unlike the other two heuristics, repeatedly applying this strategy in sequence is not advantageous because it does not consider changes of the Lagrangian multipliers.

Note, that it does not suffice to only rely on these heuristic cuts. When the heuristic cutting process stalls, it can be beneficial to solve LR(λ), since this might help to get out of a "local optimum" when the cut information obtainable from $X^1, \ldots, X^j$ is already or almost fully utilized. Finally, if no violated heuristic cut can be found, this might also be because the current solution value μ_j is in fact optimal for Problem (2), which, however, can only be proven by solving LR(λ).

To make a trade-off between adding heuristic cuts and solving the Lagrangian relaxation we set a limit on the number of heuristic cut rounds before solving the Lagrangian relaxation again. Further, we employ two criteria to evaluate the usefulness of the latest heuristic cut and switched back to the exact separation of Inequality (4) if:

1. The cut did not change the Lagrangian multipliers that minimize the cutting plane problem.
2. The efficacy, a common measure for the quality of cuts (see, e.g., Turner et al. (2023)), falls below a defined threshold tol$_e$. In iteration j the efficacy of the cut given by X^j is given as

$$e(\lambda^j, \mu^j, X^j) = \frac{LR(\lambda^j, X^j) - \mu^j}{\|\lambda^j\|}. \tag{7}$$

Figuratively, we stop if the generated cut is not separating μ^j for multiplier λ^j deep enough.

Using the described heuristic with these stopping criteria we arrive at a modified algorithm for the cutting plane based Lagrangian descent, which is outlined in Algorithm 1 for the maximum violation heuristic.

In its outer loop starting in Line 2, the algorithm performs up to $\bar{n}$ exact evaluations of multiplier λ. Once the gap between dual bound and relaxed primal bound is within a configured tolerance, Line 14 stops the algorithm.

After each multiplier evaluation, Line 6 adds up to $\bar{m}$ heuristic cuts to speed up the convergence. If an additional cut generated does not meet the efficacy tolerance tol$_e$, the resulting multipliers are not changing or the overall gap tol$_\mu$ is reached, Line 9 ends the heuristic cut generation.

We did not perform intense parameter tuning, the chosen values are mostly ad-hoc with some sensitivity tests. We limit the algorithm to $\bar{n} = 10$ iterations. Up to $\bar{m} = 10$ additional cuts could be generated. As tolerances, we ran our experiences with tol$_e = 1$ for the efficacy and tol$_\mu = 1\mathrm{e}{-6}$ for the dual bound gap.

6 Experiments and Real-world Impact

In the following section we illustrate the practical relevance of the methods described previously for the use case of the large-scale markdown price optimization system at Zalando. We report results here on a subset of the assortment for around 500,000 articles (depending on the exact

Algorithm 1: Extended cutting plane algorithm with heuristic cuts loop

```
Parameter: $\bar{n}$, $\bar{m}$, $\mathrm{tol}_\mu$, $\mathrm{tol}_\nu$, $\mathrm{tol}_e$
Let $j \leftarrow 0$, $\lambda^j \leftarrow 0$.
for $n \leftarrow 1$ to $\bar{n}$ do
    Compute $X^j$ as optimal solution of $LR(\lambda^j)$ and add the corresponding cut to Problem (3)
    Compute $\mu^j$, $\lambda^j$ as optimal solutions to Problem (3)
    $j \leftarrow j + 1$
    for $m \leftarrow 1$ to $\bar{m}$ do
        Compute a heuristic solution $X^j \in \mathcal{X}$ and add the corresponding cut to Problem (3)
        Compute $\mu^j$, $\lambda^j$ as optimal solutions to Problem (3)
        if $\frac{\min_{k \leq j} \mathrm{LR}(\lambda^k) - \mu^j}{\mu^j} < \mathrm{tol}_\mu$ or $\lambda^j = \lambda^{j-1}$ or $e(\lambda^j, \mu^j, X^j) < \mathrm{tol}_e$ then
            break
        end if
        $j \leftarrow j + 1$
    end for
    if $\frac{\min_{k \leq j} \mathrm{LR}(\lambda^k) - \mu^j}{\mu^j} < \mathrm{tol}_\mu$ then
        break
    end if
end for
```

time of the season), and for 14 countries, each with different discounts on weekly granularity until the season end of each article. Updated pricing recommendations are delivered at least twice a week, with 48 linking simplified discount rate (sDR) constraints submitted as targets (24 lower and upper bounds each). To integrate with Zalando's business processes, the turn-around time between the setting of targets and the delivery of results should usually not exceed 3 hours.

Even though considerable work has been put into improving the speed at which we can solve the article problems for one set of Lagrangian multipliers since we last reported on it (Li et al., 2022), the time to solve the Lagrangian relaxation once remains at 150-700 CPU hours (using AWS C5 instances, exact time depending on the expected lifetime of the article). Even after parallelization over very large clusters the time to complete one iteration is therefore 3-12 minutes. Given additional overhead and because of both time and budgetary constraints we are limited to 10-15 iterations, a number that is far below the number of iterations one might usually expect for a problem with this number of constraints Guignard (2003).

In the experiments, we evaluate first which of the three cut generating strategies appears to be most effective, then the influence of the quality of the cuts on the dual bound. We demonstrate that the cut heuristics are not only faster but give a better trade-off for bound improvement over time. Finally, we show how the associated commercial key performance indicators (KPI) benefit from the approach presented here.

Comparison of cut generation heuristic strategies

To explore whether the heuristics described above help improve the quality of solutions, we first tested the relative effectiveness of the different strategies. Fig. 2 illustrates the performance via a case study of a problem with problematic convergence in the baseline (no-heuristic) strategy; other instances behave similarly. We find that the maximum violation heuristic is most effective, closing the relative dual gap, $\frac{\min_{k \leq j} \mathrm{LR}(\lambda^k) - \mu^j}{\min_{k \leq j} \mathrm{LR}(\lambda^k)}$, to less than 2% after six outer loop iterations and 0.4% in the 10th iteration for a problem which did not converge after 10 iterations in the baseline (with a relative gap much larger than 100%). We observed, however, that during the first few iterations, the relaxed primal bound hardly improved, and our imposed efficacy criteria would interrupt the heuristic cut generation early and continue with a full Lagrangian step.

Applying the feasibility heuristic is also effective in generating useful additional cuts, but less so. For that heuristic, the gap only drops below 100% in the 10th iteration, with a final gap of 7.4%. Random cuts are not effective, with the results being virtually indistinguishable from not applying any heuristic. We have observed similar results on other problem instances and concluded that the maximum violation heuristic strategy is performing the best. Thus we selected it as the most promising candidate for production and further results focus only on this heuristic.

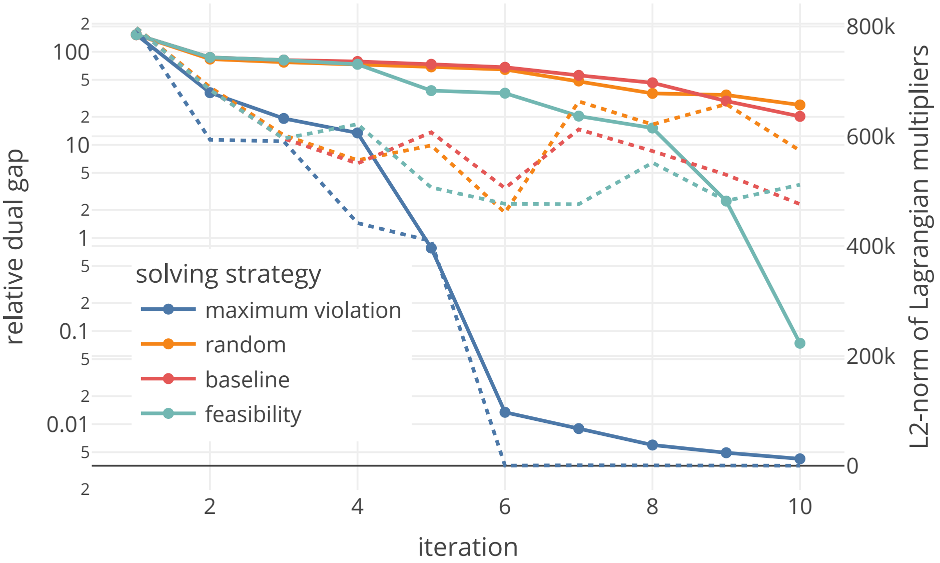


Figure 2: Solid lines and left y axis: relative dual gap of the cutting plane problem over number of outer loop cutting plane iterations. Dashed lines and right axis: L2 norm of Lagrangian multipliers λ^{j+1} selected in the cutting plane problem. Each iteration is one new solution of the Lagrangian dual. Baseline uses no heuristic to add cuts to the cutting plane problem. For details on the heuristics see section 5.

Influence on Lagrangian term and original objective

It is important to note that the contributions to the objectives, in particular for the earlier iterations are dominated by the Lagrangian term $(b - Ax)^\top \lambda$ and not the primal objective $f(x)$ as shown in more detail in Fig. 3. Both the Lagrangian term and the primal objective improve substantially after the gap closes (in this example after the sixth iteration using the maximum violation heuristic in the solving strategy).[1] At the same time the Lagrangian term remains much larger than the primal objective for the baseline solving strategy. Hence, it is plausible that for

[1] Mathematically this means that the violation of feasibility and optimality are decreased at the same time. From a business perspective, this means that the solutions from later iterations are not only more aligned with business guidelines but also more profitable – highly desirable!

the baseline, when maximizing $\mathrm{LR}(\lambda^j, X^j)$, minimizing the violations of those constraints with large multipliers λ^j will yield a larger dual than maximizing the primal objective.

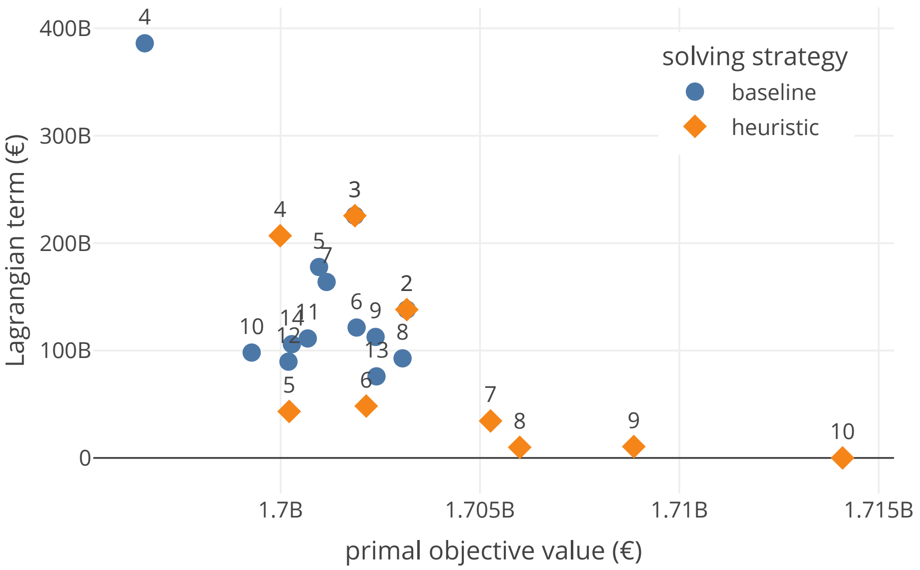


Figure 3: Primal objective values (long term profit) of outer loop iterations vs value of the Lagrangian term in the dual across iterations. Primal objective value is the value found for the (relaxed) objective $f(x)$ in iteration j. Lagrangian term is the value of $(b - AX^j)^\top \lambda^j$ of the dual solution in the iteration j. Compared are maximum violation heuristic and baseline.

We also observe in practice that we spend a lot of iterations exploring Lagrangian relaxations where at least some of the multipliers λ^j_ℓ are at the upper bound. The occurrence of the first λ^j that does not contain any multipliers at $\bar{\lambda}$ coincides with the drop in the dual gap around iteration five (Fig. 2).

Finding solutions with smaller Lagrangian multipliers is desirable if such multipliers exist that still allow for minimal or no violation of relaxed constraints, as it shifts weights in the relaxed objective towards the primal objective. Large multipliers will create extreme solutions, which are undesirable in various ways. First they will often cause the other bound to be violated (as discussed earlier, we usually set upper and lower bound constraints). And second, at those extreme solutions all prices (and costs to the primal objective) are being paid in order to not violate those linking constraints with large Lagrangian multipliers, even if smaller multipliers would be sufficient. This is exactly what we see in Fig. 3: At the same time the Lagrangian term stops dominating the Lagrangian relaxation, the primal objective improves. For us, it is of particular importance to find solutions that have both small violations of the linking constraints and a large primal objective, since we intend to use those solutions in our primal heuristic given by Problem (5). In that case, finding solutions that are close to a dual optimum, and have high primal objectives is likely to directly translate into a better final solution (after the application of the primal heuristic). Again, we conclude that from our case study, the maximum violation heuristic seems to be the most promising heuristic cut-generation approach.

Trade-off between heuristics and iterations

We know that the maximum violation heuristic adds cuts in a fashion that is similar to the application of cutting planes to the Lagrangian dual (that is iterating between the solving of the Lagrangian relaxation given some multipliers and finding the multipliers that minimize the cutting plane problem). Comparing the effectiveness of the cuts by looking at the dual gap over the number of cuts added we see that heuristic-based cuts are slightly less effective in closing the gap than cuts that solve the Lagrangian relaxation; the baseline strategy converges 23 cuts earlier (at cut 49 vs 72). (Fig. 4).

This is consistent with the assumption that solving LR(λ) in each iteration provides the best bound and the best subgradient and hence adds the most efficient cut to the Cutting Plane problem. But practically, the difference in efficiency does not appear to be very large. The objective values and gradients supplied by combining previous solutions to the Lagrangian relaxation seem to provide good enough cuts to meaningfully update the cutting plane problem.

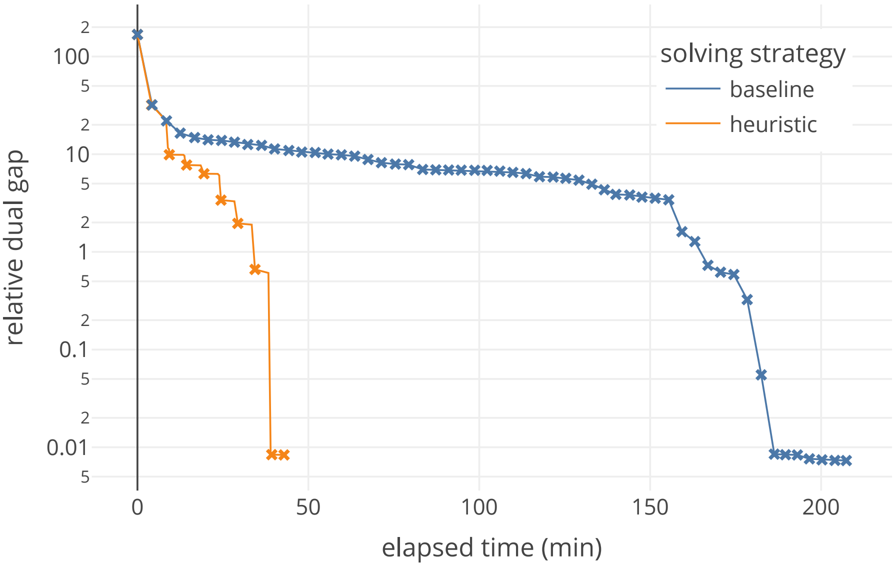


Figure 4: Relative dual gap of the cutting plane problem over time elapsed. Cuts include both those added based on the heuristic and from solving the LR. Each marker indicates the addition of a cut, and the text annotation show the number of cuts added so far. Baseline uses no heuristic to add cuts, heuristic uses the maximum violation heuristic 10 times per iteration (starting after the third iteration).

While the baseline strategy needs less iterations to converge, the time to convergence is sped up substantially when using our suggested strategy (Fig. 4). In our example, we observe a speed-up of more than 4.5 times (39 minutes vs 186 minutes). Looking across a larger set of experiments we see that the speedup is consistent in those tests (Table 1). Instead of solving the LR once, we can apply the maximum violation heuristic at least 20 times. The run-time requirement for the baseline solution are impractical in production, both because of time-constraints and due to computational costs.

We conclude that our suggested strategy of using cuts generated via the maximum violation

	avg time in seconds	standard deviation
LR solving time	256.51	53.01
Time per heuristic	12.58	3.32

Table 1: Mean and standard deviation for solving LR or heuristic for one set of multipliers, computed on 23 independent test instances using 10 iterations for a total of 230 data points.

heuristic converges much faster to similarly good solutions than the baseline strategy. Thereby it mitigates the risk of being stuck with an non-converged solution when hitting the time limit imposed by business requirements (note how in Fig. 4 convergence takes place just around the critical three hour mark).

Impact on commercial KPIs

For a fair comparison of the overall improvement that applying the maximum violation heuristic gives in a time-constrained setting, we went on to compare final solutions (after application of the primal heuristic described in Section 5) both with the maximum violation heuristic cut strategy (solving LR 10 times and heuristic 70 times) and without a heuristic cut strategy (LR 14 times).

Both strategies would take approximately equal time to solve. In this final experiment we want to investigate whether an improvement in solving the dual problem also leads to an improvement of the primal solution generated after applying our primal heuristic by solving problem (5).

Testing the improvement for 8 different dates we can see that through the application of the maximum violation heuristic in forward running simulations we achieve an uplift in all critical commercial KPIs: long-term profit (LTP, the primal objective in our formulation), first week gross merchandise volume (GMV) and the first week profit contributor 2 (PC2, which is a company accounting metric, that measures revenue minus the combined cost of sales and fulfilment). For the selected dates, we find an average weekly improvement of more than 3M€ in LTP and GMV and a 0.9M€ improvement in PC2 (Table 2). Our simulation indicates that using our suggested strategy could lead to a long term profit-increase of about 27M€ in eight weeks, which projects to 175M€ per year.

The significant enhancements observed in our simulations, combined with the minimal risk of solution deterioration (given that the heuristic cuts added are still valid, just potentially less effective), have convinced us and the responsible stakeholders to implement the maximum violation heuristic in Zalando's production environment.

Due to parallel, unrelated AB tests, we were unable test the improvements through direct comparisons, and instead opted to follow the improvements using causal impact analysis (Brodersen et al., 2015). In causal impact analysis, we select predictors that can be used to train a Bayesian structural time-series model to forecast the variables of interest. Assuming that the predictors are unaffected by the treatment we can use them to forecast what would have happened if no intervention had taken place. See also Mehrotra et al. (2020) for another usage of causal impact analysis to estimate treatment effects without relying on AB testing.

Since long term profit (the immediate objective in our markdown optimization) is not directly measurable over short observation horizons, we instead opted to focus on the improvements in PC2 and GMV we observed in our forward running simulations. We chose to use 25 weeks prior to treatment (the application of the maximum violation heuristic) for training of the time-series model and monitored the effect of treatment for 10 weeks. As covariates for the time-series model we chose the respective predictions of the markdown optimization model without the maximum violation heuristic for PC2 and GMV. Since these covariates are obtained by computing the old

Experiment	LTP	GMV	PC2
0	0.18M€	0.31M€	0.09M€
1	0.55M€	0.87M€	0.23M€
2	3.30M€	6.02M€	1.42M€
3	3.63M€	6.07M€	1.54M€
4	3.65M€	1.38M€	0.75M€
5	3.97M€	4.74M€	1.28M€
6	5.71M€	0.00M€	0.00M€
7	6.20M€	7.25M€	2.09M€
avg	3.40M€	3.33M€	0.93M€
sum	27.19M€	26.64M€	7.40M€

Table 2: Simulated improvement (in M€) of the primal solution in key KPIs for 8 start dates throughout the season. Improvements are the difference between the primal solution generated without the maximum violation heuristic, and the solution generated by applying the maximum violation heuristic 10 times per iteration. The last two rows show the average and the sum of the eight weeks

optimization without heuristic cuts and are therefore unaffected by the treatment (the use of the maximum violation heuristic in the model), this gives us a good estimator of the observed effect.

In the causal impact model (Fig. 5) we see an improvement in PC2 in the treatment period (p=0.0001, observed weekly effect: 5.8M€, expected effect 1.56M€) and an effect on GMV that is not significant, but consistent with expectations (p=0.18, observed weekly effect: 7.9M€, expected effect: 16.7M€) These findings indicate that the improvement computed in the forward running simulations are actually observable in practice, and that applying the maximum violation heuristic leads to tangible real world improvements.

7 Conclusion

In this paper, we addressed the practical challenges encountered in implementing large-scale markdown strategies using a predict-then-optimize framework. These challenges primarily revolve around the stability and efficiency of identifying close-to-optimal solutions for huge mixed integer programs. We introduced novel heuristics that enhance the performance of existing methods, facilitating the rapid identification of high-quality solutions through a Lagrangian relaxation approach. An important component was the implementation of a heuristic cut generation scheme, based on a maximum-violation measure, and its seemless integration within an existing exact separation procedure. We demonstrated the practical impact of our methodology with empirical evidence from Zalando SE's pricing systems. Our approach not only accelerates the process of finding high-quality solutions but also leads to multi-million revenue and profit increases, underscoring its commercial relevance.

References

Andersen, K., Cornuéjols, G., Li, Y., 2005. Reduce-and-split cuts: Improving the performance of mixed-integer gomory cuts. Management Science 51, 1720–1732. doi:10.1287/mnsc.1050.

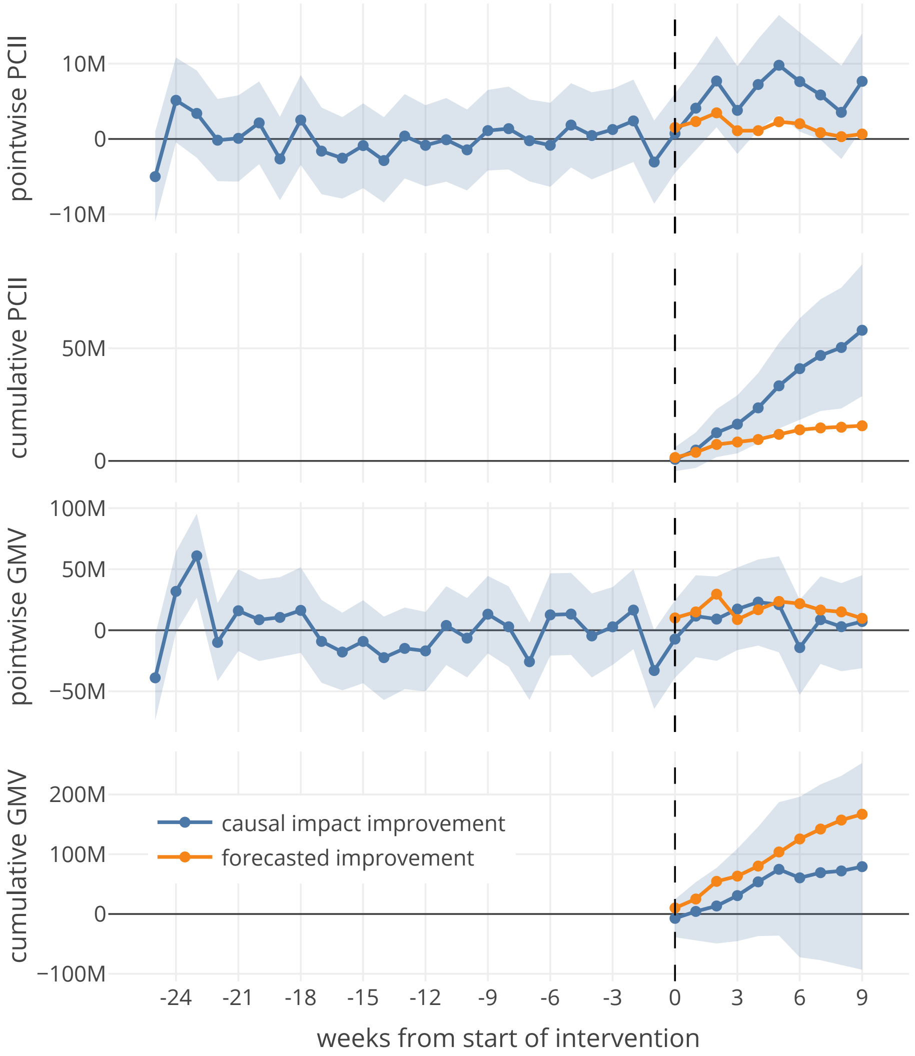


Figure 5: Causal impact analysis of the switch to the maximum violation heuristic. Analysis was performed using 25 weeks prior and 10 weeks post switching the heuristic online, using an optimizer without the heuristic as predictor and an observed PC2 and GMV as variable. We show the uplift in percent of the average weekly performance for GMV and PC2. Blue are the causal impact effects (observed - Bayesian structural time-series forecast), orange are the effects comparing optimization results without heuristic to those with heuristic.

0382.

Athey, S., Imbens, G.W., 2007. Discrete choice models with multiple unobserved choice characteristics. International Economic Review 48, 1159–1192. URL: http://www.jstor.org/stable/4542008.

Berthold, T., 2014a. Heuristic algorithms in global MINLP solvers. Ph.D. thesis. Technische Universität Berlin.

Berthold, T., 2014b. RENS – the optimal rounding. Mathematical Programming Computation 6, 33–54. doi:10.1007/s12532-013-0060-9.

Berthold, T., Lodi, A., Salvagnin, D., 2019. Ten years of feasibility pump, and counting. EURO Journal on Computational Optimization 7, 1–14.

Bonami, P., Cornuéjols, G., Dash, S., Fischetti, M., Lodi, A., 2008. Projected chvátal–gomory cuts for mixed integer linear programs. Mathematical Programming 113, 241–257.

Brodersen, K.H., Gallusser, F., Koehler, J., Remy, N., Scott, S.L., 2015. Inferring causal impact using bayesian structural time-series models. Annals of Applied Statistics 9, 247–274.

Caprara, A., Fischetti, M., 1996. {0, 1/2}-Chvátal-Gomory cuts. Mathematical Programming 74, 221–235. doi:10.1007/BF02592196.

Caro, F., Gallien, J., 2012. Clearance pricing optimization for a fast-fashion retailer. Oper. Res. 60, 1404–1422.

de Chaisemartin, C., D'Haultfœuille, X., 2020. Two-way fixed effects estimators with heterogeneous treatment effects. American Economic Review 110, 2964–96. URL: https://www.aeaweb.org/articles?id=10.1257/aer.20181169, doi:10.1257/aer.20181169.

Chen, J., Xiong, J., Chen, G., Liu, X., Yan, P., Jiang, H., 2024. Optimal instant discounts of multiple ride options at a ride-hailing aggregator. European Journal of Operational Research 314, 718–734. URL: https://www.sciencedirect.com/science/article/pii/S0377221723007944, doi:https://doi.org/10.1016/j.ejor.2023.10.019.

Deaton, A., Muellbauer, J., 1980. Economics and consumer behavior. Cambridge university press.

DeFusco, A.A., Paciorek, A., 2017. The interest rate elasticity of mortgage demand: Evidence from bunching at the conforming loan limit. American Economic Journal: Economic Policy 9, 210–240.

Eisenach, C., Patel, Y., Madeka, D., 2020. Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention. arXiv preprint arXiv:2009.14799 .

Ferreira, K.J., Lee, B.H.A., Simchi-Levi, D., 2016. Analytics for an Online Retailer: Demand Forecasting and Price Optimization. Manufacturing & Service Operations Management 18, 69–88. URL: https://ideas.repec.org/a/inm/ormsom/v18y2016i1p69-88.html, doi:10.1287/msom.2015.0561.

Fischetti, M., Lodi, A., 2003. Local branching. Mathematical Programming 98, 23–47. doi:10.1007/s10107-003-0395-5.

Fischetti, M., Salvagnin, D., 2011. A relax-and-cut framework for gomory mixed-integer cuts. Mathematical Programming Computation 3, 79–102.

Fogarty, J., 2010. The demand for beer, wine and spirits: a survey of the literature. Journal of Economic Surveys 24, 428–478.

Gamrath, G., Berthold, T., Heinz, S., Winkler, M., 2019. Structure-driven fix-and-propagate heuristics for mixed integer programming. Mathematical Programming Computation 11, 675–702. doi:10.1007/s12532-019-00159-1.

Guignard, M., 2003. Lagrangean relaxation. Top 11, 151–200. doi:10.1007/BF02579036.

Hughes, J., Knittel, C.R., Sperling, D., 2008. Evidence of a shift in the short-run price elasticity of gasoline demand. The Energy Journal 29.

Kedia, S., Jain, S., Sharma, A., 2020. Price optimization in fashion e-commerce. arXiv:2007.05216.

Kunz, M., Birr, S., Raslan, M., Ma, L., Li, Z., Gouttes, A., Koren, M., Naghibi, T., Stephan, J., Bulycheva, M., Grzeschik, M., Kekić, A., Narodovitch, M., Rasul, K., Sieber, J., Januschowski, T., 2023. Deep learning based forecasting: a case study from the online fashion industry. arXiv:2305.14406.

Li, H., Simchi-Levi, D., Sun, R., Wu, M.X., Fux, V., Gellert, T., Greiner, T., Taverna, A., 2022. Large-scale price optimization for an online fashion retailer, in: Babich, V., Birge, J.R., Hilary, G. (Eds.), Innovative Technology at the Interface of Finance and Operations: Volume II. Springer International Publishing, Cham, pp. 191–224. URL: https://doi.org/10.1007/978-3-030-81945-3_8, doi:10.1007/978-3-030-81945-3_8.

Li, H., et al., 2021. Large-scale price optimization for an online fashion retailer, in: Innovative Technology at the Interface of Finance and Operations: Volume II. Springer, pp. 191–224.

Lim, B., Arif, S., Loeff, N., Pfister, T., 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting 37, 1748–1764. URL: https://www.sciencedirect.com/science/article/pii/S0169207021000637, doi:https://doi.org/10.1016/j.ijforecast.2021.03.012.

Loh, E., Khandelwal, J., Regan, B., Little, D.A., 2022. Promotheus: An end-to-end machine learning framework for optimizing markdown in online fashion e-commerce, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. p. 3447–3457. URL: https://doi.org/10.1145/3534678.3539148, doi:10.1145/3534678.3539148.

Madeka, D., Torkkola, K., Eisenach, C., Luo, A., Foster, D.P., Kakade, S.M., 2022. Deep inventory management. arXiv:2210.03137.

Mehrotra, P., Pang, L., Gopalswamy, K., Thangali, A., Winters, T., Gupte, K., Kulkarni, D., Potnuru, S., Shastry, S., Vuyyuri, H., 2020. Price investment using prescriptive analytics and optimization in retail, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3136–3144.

Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y., 2020. N-BEATS: neural basis expansion analysis for interpretable time series forecasting, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net. URL: https://openreview.net/forum?id=r1ecqn4YwB.

Phillips, R.L., 2021. Pricing and revenue optimization. Stanford university press.

Rasul, K., Seward, C., Schuster, I., Vollgraf, R., 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, in: International Conference on Machine Learning, PMLR. pp. 8857–8868.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting 36, 1181–1191.

Schultz, D., Stephan, J., Sieber, J., Yeh, T., Kunz, M., Doupe, P., Januschowski, T., 2023. Causal forecasting for pricing. arXiv:2312.15282.

Strauss, A.K., Klein, R., Steinhardt, C., 2018. A review of choice-based revenue management: Theory and methods. European Journal of Operational Research 271, 375–387. URL: https://www.sciencedirect.com/science/article/pii/S0377221718300110, doi:https://doi.org/10.1016/j.ejor.2018.01.011.

Turner, M., Berthold, T., Besançon, M., Koch, T., 2023. Cutting plane selection with analytic centers and multiregression, in: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer. pp. 52–68.