The Integrated Size and Price Optimization Problem

August 6, 2012

Miriam Kiessling

Sascha Kurz and Jörg Rambau

Universität Bayreuth
95440 Bayreuth, Germany

Abstract

We present the Integrated Size and Price Optimization Problem (ISPO) for a fashion discounter with many branches. Based on a two-stage stochastic programming model with recourse, we develop an exact algorithm and a production-compliant heuristic that produces small optimality gaps. In a field study we show that a distribution of supply over branches and sizes based on ISPO solutions is significantly better than a one-stage optimization of the distribution ignoring the possibility of optimal pricing.

2000 Mathematics Subject Classification: Primary: 90B90, Secondary: 90B05

Key words and phrases: Supply chain, fashion discounter, lot-type, mark-down, pre-pack, stochastic optimization with recourse, field study

Supported by a grant of the *Bayerische Forschungsstiftung, Az-727-06*

1 Introduction

We want to decide on the one-time supply of seasonal apparel for a real-world fashion discounter with many branches in such a way that the expected revenue during a sales process with possible mark-downs is maximal. The supply process and the sales process of our partner have several special features. Most important among them is the following. The supply for a branch in a size can not be decided independently from the other supplies: ordering and delivery are based on pre-packed size-assortments of a product, the so-called *lots*. A *lot-type* specifies for each size the number of items in that size in the pre-pack. Another important restriction is that prices can only be marked-down for a product in all branches and all sizes, so that it is impossible to use a dynamic pricing strategy for a product in all branches and in all sizes independently.

Moreover, most fashion products are only sold once and are never offered again. Thus, historical sales data can only be used on a higher aggregation level, e.g., the average historical demand at a price in a sales week on the level. Since the average supplies per branch and size of a single product are zero, one, or two in most cases, we can expect that historical sales data will only give us very coarse information.

Thus, it seems reasonable to use an approach that takes into account forecasting inaccuracies and deviations from the normal behavior. Still we need to design a model whose stochastic parts are simple enough produce solutions fast while encompassing the indispensable operational side constraints. The mere volume of merchandise handled in a large fashion discounter (around 1000 products with, in total, around 10 million pieces per months) requires that supply calculations take no more than 15 minutes per product on average to keep up with operations. This large throughput actually implies a real-time requirement to any algorithm used in such an environment.

In this paper we try to find a balance between modeling accuracy and real-time compliance by incorporating only a few overall-success scenarios of a product into the model, but stick to point forecasts with respect to all other variabilities, like varying demands in the branches or sizes compared to each other.

1.1 Related Work

Linking of inventory and dynamic pricing decisions has been attacked in [3, 6, 9, 18]. More recent approaches consider robustness considerations [1] or game theoretic aspects, like competition and equilibria [2]. Common to those results is the optimal control approach via fluid approximation and/or the dynamic programming approach. The real-world settings of companies usually involves additional side-constraints (in our case: the restriction on the number of used lot-types) and costs (in our case: lot-type handling and opening costs) that would lead to the violation of important assumptions in optimal control and that would require very large state spaces in dynamic programming.

Dynamic pricing is a well-studied problem in the revenue management literature (see, e.g., [5, 11, 12, 17, 21] as examples). Again, complicated operational side-constraints are usually neglected in favor of a more principle study of isolated aspects. Again, some work has been done from a game theoretic point of view, like strategic customers (see, e.g., [20]).

Classical inventory management research is less related to our topic, since there most policies deal with the optimal way to replenish stock. In our environment, no replenishment is possible.

Our first steps in capturing the operations side constraints posed by the lot-based supply [13, 15] did not take pricing into account, but estimated the consequences of inventory decisions by a distance measure between supply and an estimated demand. The resulting stochastic lot-type design problem (without reference to pricing) will serve as a template for our model of the size optimization stage. Since the number of possible lot-types can be very large, we developed a branch-and-price algorithm in [15]. In this paper, however, we restrict ourselves to a managable number of applicable lot-types.

We have found no published research on field studies that apply any of the theoretical results on dynamic pricing and/or inventory control to a real-world environment and analyze the influence of the method in a planned, controlled experiment.

1.2 Our contribution

We present an inventory and dynamic pricing problem of a real-world fashion discounter with a set of operational side-constraints that has been unstudied so far. For this problem, we contribute

- a new model;
- an exact branch-and-bound algorithm for benchmarking;
- a fast heuristic for production use;
- computational results from a field experiment with a robust assessment of statistical significance.

Our combined inventory and dynamic pricing problem is new compared to already studied problems because of a combination of the following aspects:

- Seasonal items.
- No replenishing.
- No left-over items (everything is sold at some price).
- Early stock-outs are possible in some branches while others still have stock.
- Supply must be ordered and delivered in terms of lots of a limited number of lot-types (pre-packed assortments of sizes of a single product); thus, the possible inventory decisions in each branch and size are severely restricted.
- Prices must be marked-down consistently in all branches and sizes.
- Prices must be taken from a small set of possible prices.
- There are costs for handling lot-types, marginal costs for using another lot-type, and mark-down costs.

For the first time, we model this *integrated size and price optimization problem (ISPO)* as a two-stage stochastic programming problem with recourse. We consider mark-down decisions as recourse actions for small success of the product. And the real profit of a distribution of goods over branches and sizes depends on the success of the product: a branch and a size that receives too few items compared to other branches and sizes produces high opportunity costs in a high success scenario but not in a small success scenario – too few items on average can then become exactly right because of low demand. Conversely, a branch and a size that receives too many items compared to other branches produces high mark-down losses in a small success scenario but not in a high success scenario.

The two-stage setting is an approximation of a multi-stage setting: ISPO is a model to decide on inventories. It ignores the flow of information about success during the sales process (open-loop). This is acceptable because the price optimization stage in ISPO is only meant to be an estimation of the recourse cost induced by the inventory decisions. When the sales process is actually running, we plan to use the price optimization stage of ISPO with a rolling horizon to utilize weekly sales information. Thus, ISPO is

- less accurate than existing models in the price optimization stage because it estimates expected demand by an open-loop price assignment
- much more accurate than existing models in the size optimization stage because it takes into account many operational side constraints.

As a contribution to modeling real-world problems, we introduce the extended mixed integer linear programming formulation for a two-stage stochastic programming model of ISPO. For this model we design two algorithms: one exact branch-and-bound method and one heuristic method: the ping-pong heuristic. Although both methods are kind of tailor-made for our real-world problem, the underlying ideas could be used in other contexts. We state some characteristics of problems that could be attacked by similar ideas in Remarks 2 and 3.

Moreover, we performed a real-world field-study as a controlled statistical experiment (similar to a clinical study). We used in parallel an existing optimization method ("old" method) on a set of control branches and our size optimization method based on the ISPO model ("new" method) on a set of test branches. Whether a branch was assigned to be a test or a control branch is decided randomly. From this study we derived that in a five-month period we could increase the mean relative realized revenue (the mean of the total revenue divided by the maximally possible revenue) by around two percentage points (resp. more than one percentage point when only a small set of heavily cleaned up data is considered). This means big money in economies of scale.

The advantage of the controlled test set-up yields more: By using robust ranking statistics exploiting the design of the experiment, we can state that it is very unlikely (around 4 % probability) that these improvements happened by chance – and this with no assumptions on the error distributions. We have not seen any published results that investigate the significance of practical results by this (or any other) statistical method, and we consider the introduction of controlled statistical experiments into the field of retail revenue management as a contribution in its own right.

Nota bene: The "old" method with which we compared our "new" method (from this paper) is *not* the historical manual solution developed at our partner's but already a one-stage size optimization method based on the concepts in [13]; this "old" method was adopted by our partner immediately after we developed it, since it yielded obvious benefits compared to the previous, manual solution. The results from our field study, therefore, evaluate the benefit of using a stochastic two-stage model with a purely monetary objective function as opposed to a deterministic one-stage model with a non-monetary objective function based on a distance between supply and forecasted demand.

1.3 Outline of the paper

We state the ISPO formally in Section 2. Section 3 describes the extended MILP formulation of a two-stage stochastic program with recourse that we use to model the ISPO. In Section 4 we present two algorithms: one exact branch-and-bound solver of the MILP and the fast ping-pong heuristic. In Section 5 we outline the setup of our real-world field-study. Section 6 is devoted to computational results: one part underpins that ping-pong can solve real-world instances of ISPO fast with tiny optimality gaps and the other part shows the impact of using ISPO in practice. We conclude in Section 7.

2 Formal problem statement

We consider the distribution of supply over branches and sizes for a single fashion article as a two-stage optimization problem. In the *size optimization* stage (Section 2.1), we essentially decide on a *lot-type design* (see [13, 15]). In the *price optimization* stage (Section 2.2) we decide on price mark-downs during the sales process depending on the inventory induced by the first stage decisions and the overall success of the article observed after the first sales period.

We want to maximize expected profit, where profit is given by the yield during the price optimization stage minus costs for various actions in both stages (Section 2.3).

2.1 The size optimization stage

Data: Let $\mathcal{B}$ be the set of branches of our fashion discounter. Let $\mathcal{L}$ be a set of applicable *lot-types*: For a set of utilized sizes $\mathcal{S}$, a *lot-type* is a vector $(l_s)_{s \in \mathcal{S}} \in \mathbb{N}^{|\mathcal{S}|}$; it specifies the number of pieces of each size in any pre-packed lot of this lot-type.

There is an upper bound $\overline{I}$ and a lower bound $\underline{I}$ given on the total supply of the product over all branches and sizes. Moreover, there is an upper bound $\kappa \in \mathbb{N}$ on the number of lot-types used.[1]

Decisions: Consider an assignment of a unique lot-type $l(b) \in \mathcal{L}$ and an assignment of a unique multiplicity $m(b)$ to each branch $b \in \mathcal{B}$. These decisions specify that $m(b)$ lots of lot-type $l(b)$ are to be delivered to branch b.

Decision-dependent entities: The *inventory of branch b in size s* given assignments $l(b)$ and $m(b)$ is given by $I_{b,s}(l,m) = m(b)l(b)_s$. Moreover, the *total supply* resulting from $l(b)$ and $m(b)$ is given by $I(l,m) = \sum_{b \in \mathcal{B}} \sum_{s \in \mathcal{S}} I_{b,s}(l,m)$.

2.2 The price optimization stage

Data: We are given a supply $I_{b,s}$ for each branch $b \in B$ and size $s \in S$ induced by the first-stage decisions l and m. Let $k \in K = \{0, 1, \dots, k_{\max}\}$ be the index of a period, and let $\{\pi_p\}_{p \in P}$ be the set of possible prices. In each success scenario $e \in E$ we know for each price π_p, each branch b, and each size s the (fractional) mean demand $d^e_{k,p,b,s} \in \mathbb{R}_{\geq 0}$ for the product in Period k. Moreover, a start price π_0 and a salvage value $\pi_{k_{\max}}$ are given.

Realization of the demand process: The realization of the success scenarios takes place at the end of the k_{obs}th period. In all periods $0, 1, 2, \dots, k_{\text{obs}} - 1$ with yet unknown scenario the start price has to be used. Since in all periods with a choice of a price we know the success scenario we are in, only the inventory decisions of the first stage are non-anticipative. (This models the situation where the success of an article can be assessed quite well after few periods of sales.) The first k_{obs} periods could be merged to one period, but then discounting gets messy.

[1] Typical ranges of the data are $|\mathcal{B}| \approx 1000$,$|\mathcal{L}| \approx 1000$,$3 \leq |S| \leq 7$, and $2 \leq \kappa \leq 5$. The overall bounds $\underline{I}$ and $\overline{I}$ typically amount to around 10 000. We usually have $\underline{I} \approx \overline{I} - 100$. The resulting small variability for the overall supply by-passes number-theoretic problems that may occur because the total supply can only be realized as a sum of cardinalities of selected lot-types.

Decisions: For a known success scenario e we decide for each period $k \in K \setminus \{0, k_{\max}\}$ on a price index $p^e(k)$, i.e., we want to sell the product for price $\pi_{p^e(k)}$ in period k in all branches and sizes. This decision is taken at the beginning of period k_{obs}, after the realization of the success scenario.

Decision-dependent entities: Let $I^e_{0,b,s} := I_{b,s}$ for all e, b, s. Then for each period k, a selection $p(k)$ of price indices and an initial (fractional) *mean stock level*, denoted by $I^e_{k,b,s}(p)$, in that period induces (fractional) *mean sales*, denoted by $\text{sales}^e_{k,b,s}(p)$, in period k, in branch b and size s in scenario e, leading to a new mean stock level in the next period $k + 1$.

2.3 The two-stage objective

Using m lots of a lot-type l in branch b incurs a specific *lot handling cost* of $c_{l,b,m}$, e.g., a picking cost proportional to m: a lot with few pieces must be used in larger quantities and, thus, the total supply requires more picks in total.[2] For the ith used lot-type we have to pay a marginal *lot-type opening cost* of δ_i. The (fractional) *mean yield* in period k in branch b and size s induced by a price assignment $p(k)$ is given by $\text{yield}_{k,b,s}(p) = \pi_{p(k)}\text{sales}^e_{k,b,s}(p)$. Each change of prices incurs a cost of μ.

The goal is to find first-stage decisions such that for optimal second stage decisions in each scenario e the *expected profit*, which is the expectation of *total yield* minus *lot handling costs* minus *lot-type opening costs* minus *mark-down costs*, is maximal.

We call this two-stage stochastic optimization problem with recourse the *Integrated Size and Price Optimization Problem (ISPO).*

Remark 1 *Fractional inventories, sales, and demands are interpreted as approximations of expected inventories. In principle we could use many (integral) demand scenarios and integral inventory book-keeping. However, the number of necessary scenarios for such a model would be enormous. Thus, our scenarios model only the variability of the total demand induced by the overall success of the article. They do not model the variability of the actual demand with respect to periods, branches, and sizes compared to each other. These variabilities are ignored by using fractional values representing approximations of expected values. In those cases we speak of* mean values *rather than* expected values *in order to distinguish the expected values that are represented by fractional numbers from expectations that we compute explicitly over all success scenarios.*

3 Modelling

In the following, we develop an ILP formulation of the deterministic equivalent of ISPO in extended form.

For the first stage (SOP) we use binary assignment variables $x_{b,\ell,m}$ to encode the independent assignment decisions $l(b) = \ell$ and $m(b) = m$. In the second stage we introduce binary assignment variables for the independent second stage assignment decision $p^e(k)$. In order to account for the profit and the cost, we need some more

[2] A reduction of the number of picks in a high-wage country is the reason for the lot-based delivery from a low-wage country in the first place.

dependent variables. We list the complete model before we comment on the details.

$$\max - \sum_{b \in B} \sum_{\ell \in L} \sum_{m \in M} x_{b,\ell,m} \cdot c_{b,\ell,m} - \sum_{i=1}^{\kappa} \delta_i \cdot z_i \tag{1}$$

$$+ \sum_{e \in E} \mathrm{Prob}(e) \sum_{k \in K} \exp(-\rho k) \Big(\sum_{b \in B} \sum_{s \in S} r^e_{k,b,s} - \mu_k n^e_k \Big) \tag{2}$$

Size Optimization Stage (SOP):

$$\sum_{\ell \in L} \sum_{m \in M} x_{b,\ell,m} = 1 \quad \forall b \tag{3}$$

$$\sum_{m \in M} x_{b,\ell,m} \leq y_\ell \quad \forall b, \ell \tag{4}$$

$$\sum_{\ell \in L} y_\ell \leq \sum_{i=1}^{\kappa} z_i \tag{5}$$

$$z_i \leq z_{i-1} \quad \forall i \tag{6}$$

$$I_{b,s} = \sum_{\ell \in L} \sum_{m \in M} m \cdot \ell_s \cdot x_{b,\ell,m} \quad \forall b, s \tag{7}$$

$$I = \sum_{b \in B} \sum_{s \in S} I_{b,s} \tag{8}$$

$$I \in [\underline{I}, \overline{I}] \tag{9}$$

$$x_{b,\ell,m} \in \{0,1\} \quad \forall b, \ell, m \tag{10}$$

$$y_\ell \in \{0,1\} \quad \forall \ell \tag{11}$$

$$z_i \in \{0,1\} \quad \forall i \tag{12}$$

$$I_{b,s}, I \in \mathbb{Z} \quad \forall b, s. \tag{13}$$

Coupling via initial inventory:

$$I_{b,s} - v^e_{0,b,s} = 0 \quad \forall b, s, e \tag{14}$$

Price Optimization Stage (POP):

$$\sum_{p \in P} u^e_{k,p} = 1 \ \forall k, e \tag{15}$$

$$u^e_{k,0} = 1 \ \forall k < k_{\mathrm{obs}}, e \tag{16}$$

$$u^e_{k_{\max}, p_{\max}} = 1 \ \forall e \tag{17}$$

$$u^e_{k-1,p_1} + u^e_{k,p_2} \leq 1 \ \forall k, e, p_1, p_2 < p_1 \tag{18}$$

$$n^e_k \geq u^e_{k-1,p_1} + u^e_{k,p_2} - 1 \ \forall k, e, p_1, p_2 \neq p_1 \tag{19}$$

$$v^e_{k-1,b,s} - v^e_{k,b,s} = \sum_{p \in P} w^e_{k-1,b,s,p} \ \forall k, b, s, e \tag{20}$$

$$\sum_{p \in P} w^e_{k,b,s,p} \leq v^e_{k,b,s} \ \forall k, b, s, e \tag{21}$$

$$w^e_{k,b,s,p} \leq u^e_{k,p} \cdot d^e_{k,p,b,s} \ \forall k, b, s, p, e \tag{22}$$

$$r^e_{k,b,s} = \sum_{p \in P} \pi_p \cdot w^e_{k,b,s,p} \ \forall k, b, s, e \tag{23}$$

$$u^e_{k,p} \in \{0,1\} \;\forall k,p,e \tag{24}$$

$$n^e_k \in \{0,1\} \;\forall k,p,e \tag{25}$$

$$w^e_{k,b,s,p} \geq 0 \;\forall k,b,s,p,e \tag{26}$$

$$v^e_{k,b,s} \geq 0 \;\forall k,b,s,e \tag{27}$$

$$r^e_{k,b,s} \geq 0 \;\forall k,b,s,e \tag{28}$$

We first comment on the SOP stage model: We force an assignment of a lot-type and a multiplicity to each branch by Equation (3). In order to account for the opening of a lot-type, we introduce lot-type variables y_ℓ indicating whether or not lot-type ℓ is used at all and lot-type count variables z_i that take value one if and only if an ith new lot-type is used. Equation (4) guarantees that $y_\ell = 1$ whenever ℓ is assigned to at least one branch b. Inequality (5) implies that no more than κ lot-types are used. Inequality (6) enforces that $z_i = 1$ implies that the number of used lot-types is at least i. We use another dependent variable $I_{b,s}$ for the inventory in branch b and size s, and Equation (7) links this variable to the assignment decisions. The total inventory is then given by yet another dependent variable I, computed by Equation (8) and enforced to stay inside the given bounds by Inequalities (9). All independent variables have to be binary, see (10) through (12), while the dependent inventory variables are integer (13).

Next, let us have a look at the POP stage model that is linked via the start inventories $I_{b,s}$ to the SOP stage by Equations (14). Equation (15) enforces the assignment of exactly one price to each period in each scenario. That the start price and the salvage value are fixed is expressed by Equations (16) and (17). We forbid increasing prices by Equation (18). A mark-down in period k is indicated in the dependent binary variable n^e_k, which is forced to one in Inequality (19) if the price has changed compared to the previous period. The following restrictions model the dynamics of the sales process using some dependent variables. The fractional variable $v^e_{k,b,s}$ approximates the mean stock level in period k in branch b and size s in scenario e. The fractional variable $w^e_{k,b,s,p}$ measures the mean sales in period k in branch b and size s for price p in scenario e. And $r^e_{k,b,s}$ measures the mean yield in period k in branch b and size s in scenario e. (See Remark 1 for the reason why we use fractional variables here.) Equation (20) describes the change of stock levels from one period to another. Inequality (21) models that there can be no more sales than stock, and in Inequality (22) we require that, only if price p is chosen, there can be sales at price p of at most the demand at price p.

Because the objective favors larger sales, the sales variables at a price in an optimal solution will become exactly the minimum of stock and demand at that price. On the level of mean values this overestimates the mean sales; thus, this yields only an approximation. Finally, we compute by Equation (23) the yield in terms of money. In this POP stage, only the independent price assignment variables need to be binary (24). The dependent variables capturing the dynamics of mean stocks, sales, and yields are required to be nonnegative in (26) through (28).

The objective function subtracts the costs for the handling of m lots of type ℓ in branch b and the lot-type opening costs for using the first, second, . . . , ith new lot-type (1) from the expected discounted mean yields minus the expected discounted costs for mark-downs (2).

This ILP model for the deterministic equivalent of ISPO – though yet an approximation – encompasses many real-world restrictions and cost factors. Therefore, it comes as no surprise that the branch-and-bound phase of standard solvers (`cplex`[3], `scip`[4]) did not make any progress for months in all of our real-world instances. The usual real-world scale instance has 1500 branches, 5 sizes, some 2000 lot types out of which at most 5 can be used, 4 prices, optimized over a time horizon of 13 periods (usually weeks) with respect to the expectation over 3 success scenarios (success above/around/below average). And this generates a large and complicated ILP that cannot be solved by commercial-of-the-shelf methods at the time being. Thus, in the next section, we present an exact algorithm (quite fast, though not fast enough for daily operation) and a heuristic (fast enough for daily production use, and in all real-world tests with only tiny optimality gaps).

4 Algorithms for the ISPO

Since the ILP formulations presented in Section 3 cannot be solved directly we present an exact branch-and-bound algorithm in this section. The main idea is to branch on the decisions of the price optimization stage. Since the variables for the price optimization stage, see Section 3, are too finely grained, we consider more widescale decisions. A natural idea is to condense the mark-down decisions in each time period to an entire price trajectory for a given scenario $e \in E$.

We can encode the feasible mark-down strategies or price trajectories by inserting $p_{\max} - 1$ symbols for a mark-down, like e.g. $\star$, into the sequence $1, \dots, k_{\max} - 1$ (in period 0 price π_0 is fixed). An example is given by

$$\Big(1, 2, \star, 3, 4, 5, 6, 7, \star, \star, 8, 9, 10, 11, 12, \star\Big),$$

meaning that we reduce prices one time before sales period 3 and two times before sales period 8. The fourth possible reduction is delayed after the end of the whole sales period. To be more precise, the concrete prices in the different sales periods are given by:

sales period	0	1	2	3	4	5	6	7	8	9	10	11	12	13
price	π_0	π_0	π_0	π_1	π_1	π_1	π_1	π_1	π_3	π_3	π_3	π_3	π_3	π_5

Having this encoding at hand, we can state that there are exactly

$$\binom{|K| + |P| - 4}{|P| - 2} = \binom{k_{\max} + p_{\max} - 2}{p_{\max} - 1}$$

feasible mark-down strategies or price trajectories, i.e., in our example we have $\binom{16}{4} = 1820$ price trajectories for each scenario.

The exact details of the branching steps and the bounding step are outlined in Subsection 4.6. As upper bounds for the remaining size optimization stage we use both tailored combinatorial bounds, see Subsection 4.2, which can be computed efficiently, and linear relaxations. Algorithmically the efficient computation of those lower bounds is based on the fast solution of a certain subproblem, see Subsection 4.1. To remove some complications caused by this minor details one

[3] IBM ILOG CPLEX version 12.1

[4] SCIP version 2.1.0

can also assume that we solve the SOP subproblem by using the corresponding ILP formulation directly, see Subsection 4.4, without computing any cheaper bounds.

In Section 6 we present computational results for the proposed brand-and-bound algorithm.

As a heuristic, that can as well be used at the start of the branch-and-bound, we present the so-called *ping-pong* heuristic in Subsection 4.7. It will turn out that it achieves a very good solution quality while requiring only little computation time. The underlying idea is to iteratively solve the separate subproblems of the size optimization and the price optimization stage. The temporary solution of one subproblem is then taken as an input for the other subproblem. To this end we present an exact but rather easy, exact algorithm for the prize optimization stage in Subsection 4.5. To speed up ping-pong, we can use a heuristic for the SOP from [13] for the size optimization subproblem, which we recall in Subsection 4.3.

The remaining part of this section is arranged as follows. At first we present our workhorses for the solution of intrinsic subproblems in Subsections 4.1–4.5. For a first reading these can be skipped. In Subsection 4.6 we present our main branch-and-bound algorithm and in Subsection 4.7 the ping-pong heuristic.

4.1 Workhorse 1: Adjusting supplies to the total-supply constraints at minimal cost

Suppose that we want to solve the following rather general binary linear problem:

$$\min \sum_{v\in\mathcal{V}}\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} \psi(v,a,b)\cdot x_{v,a,b}$$

$$\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} x_{v,a,b} = 1 \quad \forall v\in\mathcal{V} \tag{29}$$

$$\sum_{v\in\mathcal{V}}\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} \varphi(a,b)\cdot x_{v,a,b} \in [\underline{R},\overline{R}] \tag{30}$$

$$x_{v,a,b}\in\{0,1\} \quad \forall v\in\mathcal{V}, a\in\mathcal{A}, b\in\mathcal{B}, \tag{31}$$

where

(1) $\varphi(a,b)$ is monotonously increasing in b and

(2) $\psi(v,a,b)$ is convex in b.

The continuous relaxation of this problem can be solved by a greedy approach: In the initialization phase we determine for each $v\in\mathcal{V}$ and each $a\in\mathcal{A}$ the optimal value $b_{v,a}\in\mathcal{B}$ with minimal costs $\psi(v,a,b)$ using binary search. This can be done in $O\big(|\mathcal{V}|\cdot|\mathcal{A}|\cdot\log(|\mathcal{B}|)\big)$ steps. By $v(a)$ we denote that element $a\in\mathcal{A}$ that minimizes $\psi(v,a,b_{v,a})$ and by $v(b)$ the corresponding value $b_{v,v(a)}$. With this we set $x_{v,v(a),v(b)}=1$ for all $v\in\mathcal{V}$. All other values are set to zero. If Inequality (30) is satisfied by pure chance, then the current assignment of the $x_{v,a,b}$-variables yields a globally optimal solution.

Otherwise we have to adjust the $x_{v,a,b}$ in order to satisfy the resource constraint. For brevity we only discuss the case where $\sum_{v\in\mathcal{V}}\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}}\varphi(a,b)\cdot x_{v,a,b}>\overline{R}$. The other case is analogous. Here, we iteratively have to take away resources from some of the $v \in \mathcal{V}$. To this end, we introduce *relative costs* $\Delta^-_{v,a}$ for each $v \in \mathcal{V}$ and each alternative $a \in \mathcal{A}$. Using

$$\beta_v(a) = \max\{b \in \mathcal{B} \mid \varphi(a,b) < \varphi(v(a), v(b))\}, \tag{32}$$

for a given $a \in \mathcal{A}$, the relative costs for changing the pair $\big(v(a), v(b)\big)$ to $\big(a, \beta_v(a)\big)$ are given by

$$\Delta^-_{v,a} = \frac{\psi\big(v, v(a), v(b)\big) - \psi\big(v, a, \beta_v(a)\big)}{\varphi\big(v(a), v(b)\big) - \varphi\big(a, \beta_v(a)\big)} \tag{33}$$

per resource item. If $\{b \in \mathcal{B} \mid \varphi(a,b) < \varphi(v(a), v(b))\} = \emptyset$ we set the corresponding relative costs to $\Delta^-_{v,a} = \infty$. By $\omega(v) \in \mathcal{A}$ we denote the alternative with smallest relative costs $\Delta^{-1}_v := \min\{\Delta^-_{v,a} \mid a \in \mathcal{A}\} = \Delta^-_{v,\omega(v)}$ and by $v^\star$ we denote the element in $\mathcal{V}$, where the pair $\big(\omega(v), \beta_v(\omega(v))\big)$ attains the globally smallest relative costs.

As abbreviations we use $R = \sum_{v \in \mathcal{V}} \varphi(v(a), v(b))$ and $\delta = \varphi\big(v^\star(a), v^\star(b)\big) - \varphi\big(\omega(v^\star), \beta_v(\omega(v^\star))\big)$. Due to the convexity of the target function we can state the following:

(a) If $\Delta^-_{v^\star} = \infty$, then the problem is infeasible.

(b) If $R - \delta \geq \overline{R}$, then after performing the greedily optimal replacement of the pair $\big(v^\star(a), v^\star(b)\big)$ by $\big(\omega(v^\star), \beta_v(\omega(v^\star))\big)$ the new assignments correspond to an optimal solution of our problem, where Inequality (30) is replaced by $\sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} \varphi(a,b) \cdot x_{v,a,b} \leq R - \delta$.

(c) If $R - \delta < \overline{R}$ we obtain the optimal solution of our relaxed problem with fractional variables $x_{v,a,b}$ by utilizing a suitable linear combination of the old and the new assignment.

Thus, after a finite number of iterations, depending at most linearly on the difference between the initial overall resource consumption and $\overline{R}$, we obtain the optimal solution of the problem with at most two fractional variables $x_{v,a,b}$.

We remark that is also possible to solve the integral problem by utilizing a branch-and-bound approach – we do not go into the details here.

4.2 Workhorse 2: Upper bounds for the Size Optimization Problem

In later parts of the algorithms we need a computationally cheap dual, i.e., upper bound for the SOP with a fixed scenario e and price trajectory t. We establish our first upper bound based on the integrality of the individual supply for each branch in each size, relaxing the constraints arising from a lot-based distribution.

If we supply branch b in size s with $I_{b,s}$ items, then we can compute the costs $\lambda^{e,t}_{b,s}(I_{b,s}) := \sum_{k \in K} \exp(-\rho k) r^e_{k,b,s} \in \mathbb{R}_{\geq 0}$ directly using e, t, and $I_{b,s}$ to evaluate the dependent variables $r^e_{k,b,s}$.

The supply of branch b with lot-type l in multiplicity m results in handling costs of $c_{b,l,m}$. Let $\tilde{c}_{b,s}(i) \geq 0$ be that part of the costs that can be associated with a supply of branch b in size s with i items.

By $\hat{\lambda}^{e,t}_{b,s}$ we denote the maximum value of $\lambda^{e,t}_{b,s}(I_{b,s}) - \tilde{c}_{b,s}(I_{b,s})$ for all achievable supplies $I_{b,s}$. We comment only briefly on how to compute these values fast: The most simple thing that always works, is to exhaustively enumerate the set of possible

$I_{b,s}$ (if we assume it to be finite). Once $r^e_{k,b,s}$ and $-\tilde{c}_{b,s}$ are concave functions in $I_{b,s}$ we can more sophistically compute the maximum using nested intervals.

As we have to use at least one lot-type and we assume $\delta_i \geq 0$, the costs of the objective function of ISPO that can be associated with trajectory t are bounded from above by

$$\sum_{e\in E} \operatorname{Prob}(e)\Big(-\delta_1 + \sum_{b\in B}\sum_{s\in S} \hat{\lambda}^{e,t}_{b,s}\Big). \tag{34}$$

Using the general method from Subsection 4.1, we can additionally incorporate the restrictions on the overall supply (where we assume that the convexity condition is satisfied, which is the case in our setting). Here, $\mathcal{V}$ are the pairs (b, s) of branches and sizes, $\mathcal{A}$ is an used set consisting of one element, and $\mathcal{B}$ is the set of possible supplies $I_{b,s}$ to a branch b in size s.

Another possibility to further tighten the upper bound is to incorporate the fact that the branches have to be supplied using lot-types in a certain multiplicity. So, in an initialization phase one can compute a locally best-fitting lot-type and multiplicity for each branch separately. If the number of lot-types and possible multiplicities is small enough, then this can be done simply by exhaustive enumeration. For more sophisticated methods based on a suitable parameterization of the set of applicable lot-types we refer to [15]. The restrictions on the overall supply can then be incorporated by using the algorithm from Subsection 4.1, where $\mathcal{V}$ is the set of branches B, $\mathcal{A}$ is the set of lot-types L, and $\mathcal{B}$ is the set of multiplicities M. In other words, we have relaxed the restriction to a certain number of used different lot-types and ignored the corresponding costs.

In our concrete application we have used all three mentioned upper bounds. Thus, our computational results rely on convexity; in all other cases the algorithm has to use the first bound only and will usually be slower.

4.3 Workhorse 3: A heuristic for the Size Optimization Problem

In [13] the so-called *Score-Fix-Adjust (SFA)* heuristic was proposed for the Lot-Type Design Problem (LDP). The LDP is directly related to the SOP stage of ISPO with a fixed scenario and a fixed mark-down strategy. In order to apply SFA to the SOP stage, we need to modify SFA to cope with opening and the handling costs for lot-types. Fortunately, this can be achieved by a suitable modification of the cost coefficients in an ordinary LDP: Incorporating the handling costs in the cost coefficients of an LDP is simply done by adding the handling cost $c_{b,\ell,m}$ to the cost coefficient of $x_{b,\ell,m}$. The opening costs for lot-types can be taken into account by solving for each possible number of lot-types 1 through κ an individual LDP with a prescribed numbers of used lot-types, add the corresponding opening costs to the optimal objective function, and pick the best option in hindsight.

For completeness, we briefly describe the underlying idea of the SFA heuristic for the LDP in the special form we use it stage. For each branch we determine the three locally best fitting lot-types and add a score of 100 to the best fitting lot-type, a score of 10 to the second best fitting lot-type and a score of 1 to the third best fitting lot-types. (Of course this can be generalized to the first t best fitting lot-types and different scoring schemes.) With this we have implicitly assigned a score to each lot-type $l \in L$, where most of the lot-types obtain the score zero. We can extend this scoring to the k-subsets of L by summing up the individual scores so that we implicitly get an order of the $\binom{|L|}{k}$ many feasible lot-type combinations. With this we traverse the k-subsets of L in descending order, where ties are broken arbitrarily. (The crucial observation is that this can be done without explicitly generating all such subsets beforehand.) In the fixing step we assume that the applicable lot-types are restricted to the current k-subset of L. Now we are in the situation where we can apply the algorithm from Subsection 4.1. In the initialization we start with a locally optimal assignment of lot-types and multiplicities. We choose $\mathcal{V} = B$, $\mathcal{A} = L$, and $\mathcal{B} = M$.

We remark that the SFA-heuristic reliably produces close-to-optimal solutions on real-world instances, see [13].

4.4 Workhorse 4: Exact solution of the Size Optimization Problem

For a given scenario e and a given price trajectory t, the ISPO is simplified to an LDP with modified cost coefficients. This subproblem can, e.g., be solved by utilizing the restricted version of the ILP formulation given in Section 3 – and we do this for obtaining the computational results in this paper. For more sophisticated algorithms we refer to [15], where a tailored branch-and-price algorithm is proposed that can handle millions of lot-types.

4.5 Workhorse 5: Exact solution of the Price Optimization Problem

For a given scenario e, a given mark-down strategy t, and the given initial supplies $I_{b,s}$ for all branches and sizes we can easily compute the number of sold items per branch, size, and period. Since in any reasonable setting all prices except maybe the salvage value are positive, we conclude that in any optimal solution the number of sold items is exactly the minimum of stock and demand in each period. With this all other dependent variables of the ILP formulation for the POP in Section 3 can be computed. Therefore, we can solve the POP stage by exhaustive enumeration of all possible mark-down strategies. This can be done in $O\left(|B| \cdot |S| \cdot |K| \cdot |T|\right)$ steps, which is possible in all practical situations ($|B| \approx 1000$,$3 \leq |S| \leq 7$,$|K| = 13$,$|T| = 1820$) we have encountered so far.

4.6 An exact branch-and-bound algorithm

In this subsection we propose our main algorithm – a customized branch-and-bound algorithm. We branch on maps "scenario $\mapsto$ price trajectory". A node at depth j then corresponds to all such maps with the images of the first j scenarios fixed. The leaves are the maps with fixed images for all scenarios. The cost of a leaf can be computed by solving an LDP problem from Section 4.4 (Workhorse 4), crucially depending on the method of Section 4.1 (Workhorse 1). As dual bounds we utilize the upper bounds from Subsection 4.2 (Workhorse 2). As primal bounds we employ the heuristically found solutions from Subsection 4.3 (Workhorse 3), again using the method in Secion 4.1 (Workhorse 1). In the branching step we extend a partially defined map in a node by all possible price trajectories for the next scenario.

In the following, we present the detailed implementation of the above concept. In an initialization step we compute for each scenario e and each price trajectory t

a combinatorial upper bound using the algorithms from Subsection 4.2. The bound $\Gamma(e,t)$ is saved for each pair (e,t) and possibly updated later on. Using these bounds we label the price trajectories in ascending order: $t_1^e,\dots,t_{|T|}^e$.

Next we consider the branching step at a node of Depth j, where the price trajectories $\xi_j \in T$ of the first j scenarios are already fixed. If $j < |E|$ then we consider the possible price trajectories for scenario $j+1$. We loop from $i=1$ to $i=|T|$ and consider price trajectory t_i^{j+1}. Now we compute the upper bound

$$\begin{aligned}&\sum_{h=1}^{j} \mathrm{Prob}(h)\cdot\Gamma(h,\xi_h) + \mathrm{Prob}(j+1)\cdot\Gamma(j+1,t_i^{j+1})\\ &+\sum_{h=j+2}^{|E|} \mathrm{Prob}(h)\cdot\min\big\{\Gamma(h,t)\mid t\in T\big\}\end{aligned} \tag{35}$$

for the ISPO where the first $j+1$ price trajectories are fixed to ξ_h. If this bound is smaller than the best found integral solution of the ISPO, then we can prune all price trajectories t_h^{j+1} for $h \geq i$. Otherwise we check how the bound $\Gamma(j+1,t_i^{j+1})$ was computed. If it was computed using the combinatorial relaxations from Subsection 4.2, then we compute the LP bound from the restricted ILP model, see Subsection 4.4[5], and possibly update the bound $\Gamma(j+1,t_i^{j+1})$. If the updated upper bound (35) is still to weak to prune the subtree, we fix $\xi_{j+1}=t_i^{j+1}$ and continue at the next node.[6]

In the leaves, where all price trajectories are fixed, we solve the remaining SOP, see Subsection 4.4.

4.7 The ping-pong heuristic

Since the exact algorithm is still not fast enough for daily production (see Section 6.1), we have developed a fast heuristic. The main idea is to alternatingly fix the independent variables of one stage and compute the optimal remaining variables; thereafter, the resulting independent variables of the other stage are fixed, and the remaining variables are computed optimally. And so on. To be more precise, if the independent decisions of the first stage, i.e., the supply of the branches with lot-types in a certain multiplicity – in other words: the $x_{b,l,m}$ – are given, then one can easily solve the prize optimization problem of the second stage by exhaustively enumerating all possible mark-down strategies in all scenarios separately. If for the other direction the independent decisions of the second stage, i.e., the $a_{e,t}$, are fixed, then the remaining problem reduces to the SOP stage, which is essentially an LDP with a modified cost function.

The idea now is to use the (heuristically) optimal solution of one of these two subproblems as input for the other subproblem and to iterate this until the algorithm stays at a solution. We hope that the solution that is not changed anymore is a good solution.

More specifically, we perform the following steps:

[5] Here we apply warm-start techniques and initialize the LP with a basis solution of a *similar* price trajectory within the same scenario – if available.

[6] Another possibility to improve the upper bound (35) is to replace the first sum and the central summand by the optimal target value of the LP arising from ISPO restricted to the first $j+1$ scenarios – we have not used this improvement in our computational results.

1. *Initialization:* In all scenarios we choose the mark-down strategy which produces the best combinatorial bound, see Section 4.2 (Workhorse 2).

2. Given the mark-down strategies for all scenarios we heuristically solve the remaining SOP stage with the SFA heuristic, see Section 4.3 (Workhorse 3).

3. Given the initial supply of the branches we exactly solve the prize optimization problem of the second stage, see Section 4.5 (Workhorse 5).

4. As long as the solutions of Steps 2 and 3 have not converged and the number of iterations is below a certain threshold, we proceed with Step 2.

Finally we output the best solution of the ISPO found in Step 2.

Remark 2 *The details of our branch-and-bound method are involved. However, there is one crucial property of the problem because of which the method works: Our problem has a* reversable two-stage structure. *This means: the* independent second stage *variables (in our case the maps from scenarios to price assignments) can be interpreted as* independent first stage *decisions. The independent first stage variables and all dependent variables can then be seen as second stage variables. In our setting fixing the independent decision variables of one stage does not even imply any restrictions to the feasible set of the independent variables in the other stage. We call this* reversable complete recourse. *In general, a heuristic like ping-pong is promising if fixing the independent variables from one stage leaves over a rich feasible set for the other stage, hopefully always containing improving solutions. In our case, fixing a price trajectory to a scenario does not influence the feasibility of supply. This is the case for all inventory problems where the price dependent demand can be determined a priori.*

Remark 3 *The principle of the ping-pong heuristic is similar to the principle of evolutionary algorithms, see for example [16]. The idea of evolutionary algorithms is to assign a so-called* fitness*-function to the solutions and iteratively in a* selection-step *to combine the best-solutions to get solutions with higher* fitness. *This is done until convergence. In our case the* fitness *of the supply in terms of lot-types is given by the expected revenue by the price optimization stage. By combining the local optimal supply with the local optimal mark-down strategy we possibly get a supply which results in higher revenue. One could also connect the principle of our ping-pong heuristics with the principle of bilevel programming. A bilevel program consists of an upper-level and a lower-level optimization problem. The lower-level problems considers a variable x as a parameter to compute the optimal value of a variable y while the upper-level problem obtains the optimal value of x by using the value of y computed in the lower-level problem [8]. In our case – by virtue of reversable complete recourse – we can see the size optimization stage and also the price optimization stage as both, as upper-level and as lower-level subproblems.*

5 Setup of the field study – a controlled experiment

We performed a real-world field-study as a controlled statistical experiment. On the one side we use the method currently applied by our business partner (named "old" method[7] in the following) on a set of control branches. We compared this supply strategy with the results that ISPO produced (named the "new" method in the following) on a set of test branches.

The field-study ran from end of May until end of September 2011 for 81 articles from three different commodity groups – women overgarments fashion (wof), women overgarments classic (woc) and women underwear (wu). It was necessary to select a subset of articles for the field study because the orders had already been placed in terms of lot-types, and the adaption of the supply for the test branches to the results of the new method was a far too expensive logistic operation to be carried out for each article.

Since for all advertized products, in particular for those in the field study, it is obligatory to supply each branch with at least one piece in each size, the degree of freedom in distributing the supply is severely restricted: small branches are very likely to receive the one-for-all-sizes lot, leaving fewer options for the larger branches, since the total supply is essentially fixed.

The sales process for the articles in the field study started between May 2011 and mid of June 2011 so that all articles could be observed for a time period of 15 to 17 weeks. Some further relevant properties of the used test articles are stated in Table 1.

commodity group	number of articles	number of sizes
wof	9	6
woc	9	3
wu	5	6

Table 1: Properties of the test articles.

In order to obtain statistically assessable results, we grouped the branches involved in the field study into 30 pairs according to economic key figures, like the size of the stores and revenue. Whether a branch was assigned to be a test or a control branch in such a pair was then decided randomly. In Section 6.2 we will benefit from this controlled test set-up and apply robust ranking statistics without assuming anything about the underlying error distributions.

The test branches were supplied according to close-to-optimal solutions of ISPO computed by the ping-pong heuristic. The ISPO for each article in the test selection was set-up for all branches and sizes. Since there are global constraints for the overall number of supplied items we actually *computed* the supply for *all* branches with the new method – and so did our our project partner with the old method. Our proposed supply was then *implemented* only for the *test branches*; the supply of the *control branches* (and all remaining branches) was *implemented* as computed by the old method by our project partner.

The demand $d^e_{k,p,b,s}$ was estimated based on historical sales data of articles from the same commodity group. This is highly non-trivial, and there are no publications

[7] The "old" method represents a lot-type optimization method that does not take into account the pricing stage but estimates the gain and the loss of a distribution of supply by a distance measure between the supply induced by the lot-design and the forecasted mean demand. This is not the method originally used by our business partner prior to our cooperation. The old method is essentially the Score-Fix-Adjust heuristic presented in [13] and refined in [15]. This method performed so much better than the manual solutions without optimization that it was put into operation immediately.

claiming a "best" method for this important building block. We essentially took non-parametric estimations of average values over commodity groups and interpolated values with too few observation linearly, which turned out to be more reliable than parametric estimators: the corresponding assumptions (exponentially decreasing stock, isoelasticitic dependence of the demand on the price, ...) appeared to be doubtful guesses in our environment at best.

Table 2 shows the parameter setting we used in ISPO for the field study. In order not to reveal company internals, we printed the values with respect to artificial but consistent monetary units. Important is that the handling cost $c_{b,\ell,m}$ contains a term linear in the multiplicity m: this way there is some incentive for the optimization to prefer lots that produce fewer picks in the warehouse, which is the true designation of lots in the first place. Our lot-type opening costs δ_i were estimated on the basis of a thourough cost accounting. This cost accounting also revealed that more than four lot-types can only be handled if the area for internal stock-turnover is increased substantially. The discount factor ρ was derived from an estimation of the capital binding cost. Whenever other reasons than interest rates favor faster stock-outs this can be increased. The fact that we did not account for mark-down costs μ_k just reflects the fact that at the time of the design of the experiment our partner could simply not provide a realistic value for this. (Meanwhile, we have an estimation for this as well.)

parameter	**setting**
κ	4
$c_{b,\ell,m}$	acquisition price $+ m \cdot 0.0545$ (pick cost)
δ_1	100
$\delta_i,\ i > 1$	50
E	{low, normal, high} (period-0 sales $\in [0, 10\,\%)/[10\,\%, 30\,\%]/(30\,\%, 100\,\%])$
$d_{k,p,b,s}^{\text{normal}}$	from empirical distribution and interpolation of historical sales in commodity group
$d_{k,p,b,s}^{e}$	$\alpha \times d_{k,p,b,s}^{\text{normal}}$ (α from historical sales in scenario e compared to scenario normal)
$\text{Prob}(e)$	from empirical distribution of historical sales in commodity group
k_{obs}	2 (i.e., realization of e and earliest mark-down after 2 periods)
$k_{\max}$	periods (= weeks) until end of season (article dependent)
ρ	0.000974868
$p_{\max}$	4 (five prices including start price and salvage value)
μ_k	0

Table 2: Parameter setting for the field study.

Since ISPO computes a supply for each branch and size under the assumptions that optimal (open loop) prices are chosen later on, we needed to gain control over the price optimization phase as well.

We had to decide whether we should, in the test branches,

- use an open loop price policy based on our POP model for the second stage after the revelation of the success scenario ("POP"),
- use a closed-loop pricing policy based on our POP model applied with receding horizon ("RH-POP"), or
- use whatever our partner uses for marking down prices ("manual").

Moreover, we had to decide whether or not to allow special offers and campaigns in the test and control branches.

The "POP" option would be closest to ISPO as a model, but farthest from practice: nobody would ever ignore up-to-date sales information for the mark-down decisions. The "manual" option is closest to practice if the mark-down process is not planned to change, but it leaves it open whether the applied mark-down policy is anywhere close to optimal. We chose the second option "RH-POP" because of two reasons:

- We tested the resulting closed-loop policy (in a different field study), and it performed slightly favorably compared to the manual policy that is currently in operation. Thus, the difference to the actual operations seemed not too large and it could potentially replace the mark-down system currently in use. Therefore, we concluded that it would produce results not too far from real operations.
- The POP stage of ISPO can be viewed as an estimation for the results obtained by the RH-POP. Therefore, we concluded that RH-POP would be not too far from the model assumptions either.

In order to assess better the practicability of our new method, we decided not to forbid campaigns – triggered by external reasons like new competing stores – in the test and control branches: A method the performance of which vitally relies on laboratory conditions with all exogeneous disturbances removed cannot be used in practice anyway.

We used RH-POP in a slightly refined way: For each non-negative real number we defined a success scenario: Scenario "1.0" means that the overall mean demand is as forecasted (the mean of the historical success in the commodity group). In general, Scenario "α" means that each mean demand is actually α times the forecasted mean demand. At the end of each period, we updated our estimation of α by comparison of our predicted demands (based on the old α) with the demands observed in the just completed period. Then, we used POP to compute a new open-loop price policy. Whenever the optimal price trajectory suggested a mark-down in the following two time periods, we advised our industry partner to implement exactly this mark-down. The method is reminiscent of model predictive control [14].

6 Computational results

In this section we report on extensive computational results about

- the technical performance of our algorithms in the laboratory;
- the practical performance of their solutions in the real-world field study.

6.1 Performance of the exact algorithm versus ping-pong

Table 3 shows the performances of the exact branch-and-bound algorithm compared to the performance of the ping-pong heuristic.

We ran branch-and-bound and ping-pong on many real-world instances with

- more than 1000 branches
- more than 1000 applicable lot-types, out of which at most 5 can be used in a lot-type design

- 13 periods $0, \dots, 12$
- 4 prices that can be non-increasingly set in periods 1 through 11.
- 3 scenarios for the overall success of the article, represented by demands that are $0.7, 1.0, 1.3$ times as large as a set of nominal demand values.

In the following, we present results on five such instances (results on all the other instances we tried were almost the same):

- We measured for branch-and-bound the total CPU time in hours in the column denoted by "t[h]".

  Moreover, we counted how many

  - exact computations of an ISPO with some prices fixed (see the column denoted by "#ISPO (%)")
  - exact computations of LP relaxations of an ISPO with some prices fixed (see the column denoted by "$\#\mathsf{ISPO}^{\mathsf{LP}}$ (%)")

  we needed to find and prove an optimal solution. In all other branch-and-bound nodes, it was sufficient to use the combinatorial bound coming from replacing the lot-type design restrictions to item-by-item supply. The numbers in parentheses show the percentages of the numbers of all possible branch-and-bound nodes in order to indicate how often we got away with cheap bounds only.

  Moreover, we counted the number of exact computations of an ISPO until an optimal solution was found (but not yet proved) – see the column denoted by "$\#\mathsf{ISPO}^*$". The column denoted by "t^*[h]" shows the CPU time in hours until this solution was found.

- We measured for ping-pong the CPU time in minutes until no improvement happened anymore (see the column denoted by ''t[min]"). Moreover, we counted the number of iterations with improvements in the column denoted by "#iter". Here, one iteration means one SOP and one POP computation. Finally, the column denoted by "Gap[%]" shows the relative optimality gap of the solution produced by ping-pong.

Instance	**Branch & Bound**					**Ping-Pong**		
	t[h]	#ISPO (%)	$\#\mathsf{ISPO}^{\mathsf{LP}}$ (%)	$\#\mathsf{ISPO}^*$	t^*[h]	t[min]	#iter	Gap[%]
P430204_13	13.17	14 ($<10^{-6}$)	48 (1.17)	6	7.71	10.42	2	0.028
P430206_13	15.31	24 ($<10^{-4}$)	31 (0.76)	1	0.32	3.77	1	0.015
P430207_13	43.03	80 ($<10^{-5}$)	75 (1.83)	29	3.07	9.58	1.5	0.013
P490201_13	60.78	45 ($<10^{-5}$)	177 (0.72)	23	31.67	12.62	2	0.023
P500206_13	26.79	2 ($<10^{-7}$)	13 (1.19)	1	0.94	5.47	1.5	0.000
∅	**31.82**	**33 ($<10^{-5}$)**	**68.8 (1.13)**	**12**	**8.74**	**8.37**	**1.6**	**0.016**

Table 3: Performance of the exact algorithm and the ping-pong heuristic

The results provide evidence that

- the branch-and-bound algorithm can find and prove optimal solutions for production problems in a time that makes it suitable for benchmarking purposes; it is not fast enough for daily operation, though – ;
- the combinatorial dual-bound techniques help to avoid time consuming LP computations in many nodes;
- the quality of ping-pong solutions is excellent;
- the CPU times of ping-pong are compliant to the real-time requirements of daily operation.

Thus, ping-pong could be routinely used in a field study designed as in Section 5.

6.2 Results of the field study

Our test set of articles is denoted by A. For reasons of comparability we consider for each branch the objective value of ISPO divided per merchandise value over all articles from the set A. We set the corresponding variables and parameters part from for the different articles $a \in A$ by a superscript a. Apart from that the parameters name are identical to the formulation of ISPO (Problem 3).

For each test-control pair of branches, the sums of objective function values over all articles in A were compared. That means, in particular, that expensive articles have a larger influence on the result than cheap articles. This point of view is in line with our partner's point of view.

For reasons of comparability we consider the revenues measured by the sums of objective values of ISPO divided by the maximal revenue measured by the sums of merchandise values. This means for an initial stock $I^a_{b,s}$ for the considered branch b, size s and article a and a starting price π^a_0 we compute the *relative realized objective* of the independent non-anticipative decisions

$x = \big(x_{b,\ell,m}\big)^a_{b\in B,\ell\in L,m\in M}$ as

$$RRO(x) = \frac{\text{objective achieved by } x}{\text{maximal possible objective}} =$$

$$\frac{-\sum\limits_{a\in\mathcal{A}}\sum\limits_{\ell\in L}\sum\limits_{m\in M} x^a_{b,\ell,m}\cdot c^a_{b,\ell,m} - \sum\limits_{i=1}^{\kappa}\tilde{\delta}_i\cdot z^a_i + \sum\limits_{k\in K}\exp(-\rho k)\Big(\sum\limits_{a\in\mathcal{A}}\sum\limits_{s\in S}\hat{r}^a_{k,b,s} - \tilde{\mu}^a_k\hat{n}^a_k\Big)}{-\sum_{a\in\mathcal{A}}\sum_{\ell\in L}\sum_{m\in M} x^a_{b,\ell,m}\cdot c^a_{b,\ell,m} - \sum_{i=1}^{\kappa}\tilde{\delta}_i\cdot z^a_i + \sum_{a\in\mathcal{A}}\sum_{s\in S} I^a_{b,s}\cdot\pi^a_0}. \tag{36}$$

Depending on article a, the entity z^a_i indicates that an ith lot-type was used. During the sales process, we observed $\hat{r}^a_{k,b,s}$ (the realized yield for branch b and size s in period k) for article a and $\hat{n}^a_k$ (mark-down in period k – yes or no).

Since we only consider a subset of branches we have to take into account that pick costs, costs for additional lot types, and fixed mark-down costs must be scaled with respect to the number of considered branches. This way, we get a marginal cost $\tilde{\delta}_i$ for the ith selected lot-type and mark-down costs $\tilde{\mu}^a_k$ for period k. (For a complete notational reference see Section 3).

The relative realized revenues are shown in Table 4 for each pair at the second and third column.

test-control-pair	RRO_{test}	RRO_{control}	$RRO_{\text{test}} - RRO_{\text{control}}$	**signed rank**
1	0.6333	0.6214	0.0119	2
2	0.6764	0.6080	0.0683	19
3	0.5919	0.6072	-0.0154	-5
4	0.6056	0.5898	0.0159	6
5	0.6637	0.5663	0.0974	26
6	0.6228	0.6031	0.0197	8
7	0.6377	0.6500	-0.0123	-3
8	0.5832	0.5845	-0.0013	-1
9	0.5968	0.5731	0.0237	11
10	0.5372	0.6276	-0.0904	-23
11	0.5651	0.5489	0.0163	7
12	0.5333	0.5904	-0.0571	-18
13	0.5782	0.5570	0.0212	9
14	0.6381	0.4940	0.1441	28
15	0.5054	0.5845	-0.0791	-21
16	0.5927	0.4993	0.0934	25
17	0.5872	0.4943	0.0929	24
18	0.6078	0.5691	0.0388	16
19	0.5762	0.6476	-0.0714	-20
20	0.5682	0.5323	0.0359	14
21	0.5133	0.4250	0.0883	22
22	0.5272	0.5547	-0.0275	-12
23	0.4015	0.5942	-0.1926	-30
24	0.4628	0.4860	-0.0232	-10
25	0.5168	0.4646	0.0522	17
26	0.5843	0.4621	0.1222	27
27	0.5658	0.4137	0.1521	29
28	0.4989	0.4608	0.0380	15
29	0.5466	0.5607	-0.0141	-4
30	0.5593	0.5272	0.0320	13
∅	0.5692	0.5499	0.0193	5.7

Table 4: RROs for the test-control-pairs – all 81 articles

Wee see that on average, the use of the new method gains almost two percentage points compared to the old method.

In the following we use the controlled setup of the field study to argue that the results are statistically significant with a prescribed significance level of 5 % with no assumptions for the error distributions (see, e.g., [10] for general information on hypothesis testing).

We apply Wilcoxon signed-rank test from statistics [19]. This test is applied for statistical experiments for two related ordinal samples where no underlying distribution can be assumed. It is an alternative to the Student's t-test, which is applied for two related ordinal samples under the assumption that the observations are normally distributed.

The procedure is as follows: The differences of the observations, here $RRO_{\text{test}} - RRO_{\text{control}}$ – at the fourth column of Table 4 are ordered increasingly according to their absolute values. The ordering implies the corresponding rank for the test-control-pair. Moreover, the sign of $RRO_{\text{test}} - RRO_{\text{control}}$ is assigned to the rank. If the test branch won, than the rank has positive sign, otherwise negative, see the fifth column of Table 4. The rank sum is the sum of all ranks with positive sign. To check significance in terms of a better performance of the test branches, we compute the probability that this or a higher rank-sum is observed by pure chance.

Our null-hypothesis is: using the new method does not improve operations systematically. That is, it does not increase the probability to obtain a better objective function value in practice.

The motivation for this test is: If the null-hypothesis is true, the signed-rank sum would lead to a rank-sum close to 0 with no systematic positive deviation.

More specifically: With a predefined significance level of α we can reject our null hypothesis "test branches not systematically better than control branches" whenever we observe rank-sum k and $P_n(X \geq k) < \alpha$ where n is the number of test-control-pairs.

For the data from Table 4 we get a rank-sum of 318. The probability for getting an equal or higher rank-sum is $P_{30}(X \geq 318) \approx 4.02\%$.

Thus, we can reject the null hypothesis with a significance level of 5 %. Consequently, the test branches performed for the whole test set of 81 articles significantly better than the control branches.

However, we could observe that there were some operational anomalities like failed price cuts in the control branches. In order to estimate the influence of the new method in the most conservative fashion, we removed all articles which may have been affected by systematic disturbances of operations. This led to a second set of articles A' with only 23 articles remaining.

The particular RROs are stated in Table 5. We see that in the case of heavily cleaned-up data the RRO for the test branches is still more than 1.5 percentage points higher than in the control branches. We repeated Wilcoxon signed-rank test for this smaller test set. Wilcoxon signed-rank test now yields a rank-sum of 271, which leads to a probability of $P_{30}(X \geq 271) = 22\%$ that a better performance of the test-branches was observed by pure chance. Thus, for the heavily cleaned-up data we still observe a relevant effect (1.5 percentage points improvement) whose observation can no longer be testified as significant. This is essentially caused by the fact that for such a small (but relevant) effect the sample set A' is simply no longer large enough to prove significance. Still, the probability for a systematic improvement is much larger than the probability that the observed effect was caused

test-control-pair	RRO_{test}	RRO_{control}	$RRO_{\text{test}} - RRO_{\text{control}}$	**signed rank**
1	0.4215	0.6673	-0.2458	-28
2	0.5874	0.4758	0.1116	18
3	0.6572	0.4865	0.1708	25
4	0.5948	0.4773	0.1175	21
5	0.5491	0.4153	0.1338	24
6	0.5799	0.5117	0.0682	13
7	0.4833	0.5454	-0.0621	-12
8	0.4648	0.5124	-0.0476	-9
9	0.5051	0.4923	0.0128	2
10	0.4933	0.6094	-0.1162	-19
11	0.4926	0.4998	-0.0071	-1
12	0.4205	0.4706	-0.0501	-10
13	0.4352	0.3746	0.0607	11
14	0.7046	0.2860	0.4186	30
15	0.4547	0.5281	-0.0734	-14
16	0.5146	0.3846	0.1300	22
17	0.5285	0.4247	0.1038	17
18	0.4802	0.5081	-0.0279	-3
19	0.3562	0.4865	-0.1303	-23
20	0.4119	0.4496	-0.0377	-5
21	0.2195	0.2577	-0.0382	-6
22	0.4274	0.5437	-0.1163	-20
23	0.2262	0.6415	-0.4153	-29
24	0.4006	0.3252	0.0754	16
25	0.3779	0.4244	-0.0465	-8
26	0.4759	0.4008	0.0750	15
27	0.5926	0.3971	0.1955	26
28	0.4458	0.4116	0.0342	4
29	0.4540	0.4985	-0.0445	-7
30	0.5278	0.3050	0.2228	27
∅	0.4761	0.4604	0.0157	2.57

Table 5: RROs for the test-control-pairs – heavily cleaned-up data, 23 articles

by pure chance.

So far, we assessed the quality of the decisions of the various methods on the basis of our objective function that was carefully enginieered together with our partner. Yet, it is interesting to see that the new two-stage method outperforms the old method in some very important criteria at the same time. In Table 6.2 we list average RRO, relative gross yields, and relative sales for all test-control-pairs. For both revenue and gross yield we see improvements by the new method. In contrast to this, the number of sales is only minimally smaller for the new method.

Now, which decisions have been taken differently by the new method? On the heaviliy cleaned-up data set of 23 articles, the new price optimization suggested alltogether 16 mark-downs in the test branches, while the manual strategy in the control branches led to 18 mark-downs on the same set. This difference may be caused by the fact that the new method tries to balance the increase in sales against the decrease in the yield per piece more thoroughly.

Table 7 shows the differences in the lot-type designs of the new and the old method for the 23 remaining articles.[8] The most obvious effect is that the number

[8] Since the lot-type design of the control branches had to be reconstructed from in this respect incomplete transaction data, the multiplicities for the control branches do not always add up to 30. The lot-types are reliable, though.

sample	relative realized objective	gross yield	sales
test	0.4761	0.6829	0.7951
control	0.4604	0.6744	0.8021

Table 6: Alternative performance metrics, heavily cleaned-up data.

of different lot-types used is usually smaller for the new method than for the old method. Since the old method tries to approximate a fractional demand as closely as possible by a supply distribution on the basis of suitable lot-types, it will usually use as many lot-types as possible, even if the improvements of a new lot-type are small. The goal of the new method is not to meet the demand as closely as possible but to earn as much money as possible. Obviously, an additional lot-type is not always justified by higher predicted profits in ISPO. Consequently, ISPO does not suggest to use such a new lot-type. In the table we clearly see that lot-type $(1, \ldots, 1)$ is very often used. This is the result of the rule that each branch has to receive at least one piece in every size – a fact that reduces the potential for improvement and should be taken into account when the effect (1.5 to 2 percentage points improvement) of using the new method is assessed.

no.	lots delivered to test branches by new method	lots delivered to control branches by old method
1	4(2,2,3,4,3,3),19(1,1,1,1,1,1),7(1,1,2,2,2,2)	13(1,1,1,1,1,1),7(1,1,1,2,2,1),5(1,1,2,2,3,2),3(2,3,3,4,4,3)
2	4(2,2,3,4,3,3),19(1,1,1,1,1,1),7(1,1,2,2,2,2)	13(1,1,1,1,1,1),7(1,1,1,2,2,1),5(1,1,2,2,3,2),3(2,3,3,4,4,3)
3	5(2,2,3,4,3,3),18 (1,1,1,1,1,1),7(1,1,2,2,2,2)	15(1,1,1,1,1,1),7(1,1,1,2,2,1),3(2,3,3,4,4,3),3(1,1,2,2,3,2)
4	5(2,2,3,4,3,3),18(1,1,1,1,1,1),7(1,1,2,2,2,2)	12(1,1,1,1,1,1),8(1,1,1,2,2,1),3(1,1,2,2,3,2),3(2,3,3,4,4,3)
5	5(2,2,3,4,3,3),18(1,1,1,1,1,1),7(1,1,2,2,2,2)	13(1,1,1,1,1,1),7(1,1,1,2,2,1),4(1,1,2,2,3,2),4(2,3,3,4,4,3)
6	5(2,2,3,4,3,3),18(1,1,1,1,1,1),7(1,1,2,2,2,2)	14(1,1,1,1,1,1),7(1,1,1,2,2,1),6(1,1,2,2,3,2),3(2,3,3,4,4,3)
7	5(2,2,3,4,3,3),18(1,1,1,1,1,1),7(1,1,2,2,2,2)	14(1,1,1,1,1,1),7(1,1,1,2,2,1),6(1,1,2,2,3,2),3(2,3,3,4,4,3)
8	5(2,2,3,4,3,3),18(1,1,1,1,1,1),7(1,1,2,2,2,2)	12(1,1,1,1,1,1),7(1,1,1,2,2,1),7(1,1,2,2,3,2),3(2,3,3,4,4,3)
10	13(1,1,1,2,2,2),17(1,1,1,1,1,1)	6(2,2,2,3,4,4),8(1,1,1,2,3,3),12(1,1,1,2,2,2),4(1,1,1,1,1,1)
11	13(1,1,1,2,2,2),17(1,1,1,1,1,1)	6(2,2,2,3,4,4),8(1,1,1,2,3,3),12(1,1,1,2,2,2),4(1,1,1,1,1,1)
12	13(1,1,1,2,2,2),17(1,1,1,1,1,1)	6(2,2,2,3,4,4),8(1,1,1,2,3,3),12(1,1,1,2,2,2),4(1,1,1,1,1,1)
14	13(1,1,1,2,2,2),17(1,1,1,1,1,1)	9(1,1,2,2,2,2),9(1,1,1,1,1,1),3(1,1,2,2,1,1),6(1,1,1,1,2,2)
16	10(1,1,1,2,2,2),7(1,1,2,2,2,2),13(1,1,1,1,1,1)	14(1,1,2,2,3,3),5(2,2,3,4,4,4),11(1,1,1,2,2,2)
17	10(1,1,1,2,2,2),7(1,1,2,2,2,2),13(1,1,1,1,1,1)	14(1,1,2,2,3,3),5(2,2,3,4,4,4),11(1,1,1,2,2,2)
18	10(1,1,1,2,2,2),7(1,1,2,2,2,2),13(1,1,1,1,1,1)	14(1,1,2,2,3,3),5(2,2,3,4,4,4),11(1,1,1,2,2,2)
19	18(3,2,1),12(2,1,1)	10(4,2,1),19(3,2,1)
20	8(1,3,2),22(1,2,1)	10(1,2,1),6(2,4,3),11(1,3,2),2(1,1,1)
21	8(1,3,2),22(1,2,1)	22(1,2,1),6(1,1,1),2(1,3,1)
22	7(2,4,3),11(1,2,1),4(2,3,2),8(1,3,2)	16(1,2,1),7(2,4,3),3(1,3,2),1(1,2,2)
23	18(3,2,1),12(2,1,1)	1(2,1,1),9(4,2,1),18(3,2,1),1(1,1,1)

Table 7: Supply for the test and control branches in terms of lots.

In Tables 8 and 9 we show how well ISPO predicted the expected function values and the expected sales, resp. While the prediction quality of the expected function values seems unsatisfactory, we get that the prediction of sales is quite good. That sales can be predicted well is more an indication for the fact that essentially everything is sold anyway. What matters more is how much money can be earned by these sales. And this in turn indicates that it is vital to estimate the return when it comes to deciding about the distribution of supply. Although our predictions are presumably biased (we usually predict better function values than the realized ones), the volatility even in one commodity group is very high (expressed

	com. group	no.	article	predicted	realized	gap
	wof	1	1597437	527.96	318.64	-0.3965
	wof	2	1597438	527.96	285.88	-0.4585
	wof	3	1597791	900.39	490.29	-0.4555
	wof	4	1598323	900.39	603.34	-0.3299
	wof	5	1598324	900.39	533.66	-0.4073
	wof	6	1599002	391.20	482.88	0.2343
	wof	7	1599007	391.20	415.96	0.0633
	wof	8	1599843	700.42	656.63	-0.0625
	wof	9	1599850	700.42	521.56	-0.2554
	woc	10	1593027	620.24	440.91	-0.2891
	woc	11	1593028	620.24	435.63	-0.2976
	woc	12	1593029	620.24	497.15	-0.1984
	woc	13	1593055	957.87	666.63	-0.3041
	woc	14	1593056	957.87	545.82	-0.4302
	woc	15	1593057	957.87	622.68	-0.3499
	woc	16	1593079	631.84	680.98	0.0778
	woc	17	1593080	631.84	664.36	0.0515
	woc	18	1593081	631.84	651.06	0.0304
	wu	19	1595383	393.90	292.29	-0.2580
	wu	20	1597776	624.91	414.50	-0.3367
	wu	21	1597803	262.52	297.26	0.1324
	wu	22	1598044	371.35	508.94	0.3705
	wu	23	1599151	421.82	364.31	-0.1363
∅						-0.1742
sd						0.2388

Table 8: Comparison of objective function values – predicted by ISPO versus realized.

by the standard deviation): a gap of zero is still inside the interval "average minus standard deviation" through "average plus standard deviation". Yet, we will try to reduce the bias of the prediction in the future by comparing realizations and predictions more carefully.

7 Conclusion and future work

We introduced the integrated size and price optimization problem ISPO, which is a two-stage stochastic optimization problem with recourse to optimize the distribution of goods among branches and sizes for a fashion discounter. We presented an MILP formulation of the deterministic equivalent in extensive form. This model, however, could not be solved for real-world instances by commercial MILP software of the shelf. We therefore suggested one exact branch-and-bound algorithm for benchmarking and a ping-pong heuristic for daily production use. In computational experiments on real-world data we showed that the optimality gap of ping-pong is usually tiny. In a five-month field study we applied ISPO in practice to distribute produces over branches and sizes and observed the sales process thereafter. We obtained an improvement for the realized relative objective of more than 1.5 percentage points compared to a one-stage lot optimization model. Because the field study was designed as a controlled statistical experiment, we could show that (for the complete set of the test articles) it is very unlikely that an improvement

	com. group	no.	article	predicted	realized	gap
	wof	1	1597437	232.295	194	-0.1649
	wof	2	1597438	232.295	177	-0.2380
	wof	3	1597791	231.724	180	-0.2232
	wof	4	1598323	231.724	210	-0.0937
	wof	5	1598324	231.724	198	-0.1455
	wof	6	1599002	227.202	235	0.0343
	wof	7	1599007	227.202	214	-0.0581
	wof	8	1599843	225.305	253	0.1229
	wof	9	1599850	225.305	226	0.0031
	woc	10	1593027	204.52	203	-0.0074
	woc	11	1593028	204.52	206	0.0072
	woc	12	1593029	204.52	207	0.0121
	woc	13	1593055	199.606	221	0.1072
	woc	14	1593056	199.606	204	0.0220
	woc	15	1593057	199.606	218	0.0922
	woc	16	1593079	225.639	251	0.1124
	woc	17	1593080	225.639	237	0.0504
	woc	18	1593081	225.639	235	0.0415
	wu	19	1595383	138.836	125	-0.0997
	wu	20	1597776	122.689	95	-0.2257
	wu	21	1597803	124.373	123	-0.0110
	wu	22	1598044	191.482	213	0.1124
	wu	23	1599151	138.836	130	-0.0636
∅						-0.0267
sd						0.1133

Table 9: Comparison of sales – predicted by ISPO versus realized.

happened by pure chance.

In order to be able to cope with more applicable lot-types, it would be very interesting to generalize the branch-and-price algorithm for the SLDP (size optimization only by solving the stochastic lot-type design problem) in [15] to the ISPO. The ping-pong heuristic computes solutions to first and second stage seperately with the variables of the other stage fixed; thus, at least the ping-pong heuristic should also work with many applicable lot-types.

Similarly, aggregating price selections to complete markdown strategies like in Section 4 could possible be used to generate a tighter ILP formulation with an independent decision variable for each markdown strategy. The more relevant advantage of such a formulation, however, is the following. Since our problem can be formulated as a two stage stochastic integer linear program, one might apply corresponding general algorithms from that area, see, e.g., [4] for an overview. A quite common algorithm is the so-called L-shaped method (stochastic Benders decomposition of the second stage). A necessary condition for the L-shaped method is, that the target function $Q(x, e)$ of the second stage is concave and continuous (for maximization), which is often violated in the presence of second-stage integrality constraints. However, if variables for complete price trajectories are used instead of periodical price assignment variables, then the second stage is mathematically more well-behaved. This might be a promising direction for further research.

The most important question posed by this work is, however: Can the demand forecasts be improved by statistical methods, with which many parameters can be estimated well by few observations? This points into the direction of support vector machines [7]. It would be interesting to learn whether or not the practical impact of our optimization results will improve if such more sophisticated forecasting methods are used in practice.

References

[1] E. Adida and G. Perakis. A robust optimization approach to dynamic pricing and inventory control with no backorders. *Mathematical Programming*, 107:97–129, 2006. 10.1007/s10107-005-0681-5.

[2] E. Adida and G. Perakis. Dynamic pricing and inventory control: Uncertainty and competition. *Oper. Res.*, pages 289–302, 2010.

[3] D. Bertsimas and S. de Boer. Dynamic pricing and inventory control for multiple products. *Journal of Revenue & Pricing Management (JRPM)*, 3:303–319, 2005.

[4] J. R. Birge and F. Louveaux. *Introduction to stochastic programming. 2nd ed.* Springer Series in Operations Research and Financial Engineering. New York, NY: Springer. xxv, 485 p., 2011.

[5] G. Bitran and R. Caldentey. An overview of pricing models for revenue management. *Manufacturing & Service Operations Management (MSOM)*, 5(3):203–229, 2003.

[6] L. M. A. Chan, Z. J. M. Shen, D. Simchi-Levi, and J. Swann. Coordination of pricing and inventory decisions: A survey and classification. In S. D. Wu D. Simchi-Levi and Z. J. M. Shen, editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, chapter 14, pages 335–392. Kluwer Academic Publishers, 2004.

[7] Andreas Christmann and Ingo Steinwart. *Support Vector Machines.* Information Science and Statistics. Springer, 2008.

[8] Antonio J. Conejo, Enrique Castillo, Roberto Minguez, and Raquel Garcia-Bertrand. *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications.* Springer, softcover reprint of hardcover 1st ed. 2006 edition, 11 2010.

[9] A. Federgruen and A. Heching. Combined pricing and inventory control under uncertainty. *Oper. Res.*, 47(3):454–475, 1999.

[10] D. Freedman, R. Pisani, and R. Purves. *Statistics, 4th Edition.* W. W. Norton & Company, 4th edition, 2 2007.

[11] G. Gallego and G. van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Manage. Sci.*, 40:999–1020, August 1994.

[12] G. Gallego and G. van Ryzin. A multiproduct dynamic pricing problem and its applications to network yield management. *Oper. Res.*, 45(1):24–41, 1997.

[13] C. Gaul, S. Kurz, and J. Rambau. On the lot-type design problem. *Optim. Methods Softw.*, 25(2):217–227, 2010.

[14] Lars Grüne and Jürgen Pannek. *Nonlinear Model Predictive Control*. Communications and Control Engineering. Springer, 1st edition, 2011.

[15] M. Kießling, S. Kurz, and J. Rambau. An exact column-generation approach for the lot-type design problem. Preprint, Universität Bayreuth, 2012.

[16] Kaisa M. Miettinen. *Evolutionary Algorithms in Engineering & Computer Science*. John Wiley and sons LTD, 1999.

[17] A. L. Montgomery. The implementation challenge of pricing decision support systems for retail managers. *Appl. Stoch. Models Bus. Ind.*, 21(4-5):367–378, 2005.

[18] S. Netessine. Dynamic pricing of inventory/capacity with infrequent price changes. *Eur. J. Oper. Res.*, 174(1):553–580, 2006.

[19] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[20] R. Yin, Y. Aviv, A. Pazgal, and C. S. Tang. Optimal markdown pricing: Implications of inventory display formats in the presence of strategic customers. *Manage. Sci.*, 55:1391–1408, August 2009.

[21] W. Zhao and Y.-S. Zheng. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science*, 46(3):375–388, 2000.