Regionalized Assortment Planning for Multiple Chain Stores: Complexity, Approximability, and Solution Methods

Hans Corsten, Michael Hopf, Benedikt Kasper, Clemens Thielen

Abstract In retail, assortment planning refers to selecting a subset of products to offer that maximizes profit. Assortments can be planned for a single store or a retailer with multiple chain stores where demand varies between stores. In this paper, we assume that a retailer with a multitude of stores wants to specify her offered assortment. To suit all local preferences, regionalization and store-level assortment optimization are widely used in practice and lead to competitive advantages. When selecting regionalized assortments, a trade-off between expensive, customized assortments in every store and inexpensive, identical assortments in all stores that neglect demand variation is preferable.

We formulate a stylized model for the regionalized assortment planning problem (APP) with capacity constraints and given demand. In our approach, a common assortment that is supplemented by regionalized products is selected. While products in the common assortment are offered in all stores, products in the local assortments are customized and vary from store to store.

Concerning the computational complexity, we show that the APP is strongly NP-hard. The core of this hardness result lies in the selection of the common assortment. We formulate the APP as an integer program and provide algorithms and methods for obtaining approximate solutions and solving large-scale instances.

Lastly, we perform computational experiments to analyze the benefits of regionalized assortment planning depending on the variation in customer demands between stores.

Michael Hopf, Clemens Thielen
University of Kaiserslautern, Department of Mathematics
Paul-Ehrlich-Str. 14, D-67663 Kaiserslautern, Germany, e-mail: {hopf,thielen}@mathematik.uni-kl.de

Hans Corsten, Benedikt Kasper
University of Kaiserslautern, Department of Business Studies and Economics
Gottlieb-Daimler-Str. 42, D-67663 Kaiserslautern, Germany, e-mail: {benedikt.kasper,corsten}@wiwi.uni-kl.de

1 Introduction

A retailer with a multitude of stores has two basic strategies for specifying her offered assortment:

- Every store has a customized assortment. Here, all demand differences can be considered, but customized assortments are expensive to maintain.
- Assortments in all stores are the same. Here, the assortments may not be optimized to suit all local preferences, but with a single assortment, economies of scale and a recognition value can be generated.

In this paper, we analyze the benefit of mixed strategies, i.e., the selection of a *common assortment* that is supplemented by regionalized products. Thus, products in the common assortment are offered in all stores, while products in the local assortments are customized and vary from store to store.

1.1 Previous Work

The assortment planning problem (APP) is considered in both operations research and retail literature in various settings. Nevertheless, researchers always consider one of the basic strategies described above. For extensive reviews, see [10, 14, 15, 20]. Generally, most researchers take shelf space [3], inventory [7, 9], or pricing decisions [25] into account. If a product is not available, customers may substitute it by an offered one. Thus, demand depends on the offered assortment.

Usually, choice models are used to estimate demand based on actual customer behavior. A review of different choice modeling approaches can be found in [14]. Common approaches are multinomial logit (MNL) models [8, 19], nested logit models [12], locational choice models [7], exogenous demand models [13], and generalizations thereof [5, 24]. Other approaches for estimating demand are described, e.g., in [1, 2], where four factors that influence demand are identified, in [23], where demand depends on displayed inventory, and in [6, 18], where attribute-based approaches are proposed.

In [22], the authors show that a greedy algorithm is optimal for an unconstrained assortment planning problem under an MNL model. In [19], the researchers give an example where a greedy heuristic is suboptimal for a capacitated problem and develop a polynomial-time algorithm. [16] proposes an assortment selection heuristic for a problem with given demand. In [3], the authors formulate a sequential assortment and shelf space allocation procedure when gross margins are given. [17] proposes a robust knapsack model for the assortment planning problem with given profit. Our approach follows these assumptions: we formulate a stylized model for the regionalized assortment planning problem with given demand.

Regionalization and store-level assortment optimization lead to competitive advantages and are widely used in practice (see, e.g., [6, 8, 11]). In [14], the authors describe this as follows: Chain store management dictates a portion of the assortment that is carried in all stores, while the remainder is chosen to satisfy local customer preferences. Surprisingly, very little research is done in this context.

In [6], the authors consider a similar problem: They characterize an algorithm for assortment planning that allows a maximum number L of individual assortments (where L is chosen between 1 and the number of stores I). To achieve this, they construct L store clusters and identify optimal assortments. While this leads to a reduced number of assortments, these assortments are independent of each other. With independent assortments, economies of scale cannot be generated. [21] allocates products to assortment modules. Then, these modules are assigned to stores. Hereby, a trade-off between standardization and individualization is possible. But, as in [6], a recognition value and economies of scale cannot be generated. [18] develops a model where only a fraction of products can change between two periods. If they use a chain-wide assortment as a starting point and apply their model once at every individual store, custom assortments that differ in a fixed fraction from the original assortment can be generated.

1.2 Our Contribution

We propose an alternative solution method that reflects industry practice. Items for the common assortment and items for the local assortments are selected simultaneously in order to maximize the total profit. We show that this problem is strongly NP-hard and present a heuristic that is able to tackle large-scale instances that cannot be solved within a reasonable amount of time by applying a commercial solver to a standard integer programming formulation. Moreover, we evaluate the quality of our algorithm in several computational experiments.

2 Formulation of the APP as an Integer Program

In this section, we formulate a stylized model for the APP that is obtained by simplifying the original problem using some reasonable assumptions (see also [4]).

Assumption 1 *We develop our model using the following assumptions:*

- *The assortment consists of standardized products, offered at standardized shelf space (i.e., all products have unit size).*
- *Every store has the same capacity.*
- *Demand can be estimated for every product and every store.*
- *Products that are assigned to the common assortment are offered in every store, while products in a local assortment are offered only in this particular store.*

The most restrictive assumption is that all products have unit size. However, we can view the unit size as a standardized area (e.g., $1m^2$) that is occupied by each item. Then, an item corresponds to the number of units of a particular product that fit into this area (e.g., 1000 pencils, 4 toasters). The assumption that every store has the same capacity can be loosened (see Section 4).

Now, we describe the APP as a variant of the multiple knapsack problem. We are given m bins (stores) with size K (capacity of the store) in which we want to pack unit size items (products). In total, there are n different items. With each item j, we associate $m+1$ different profits $w_j \geq 0$ and $v_{jk} \geq 0$ for $k = 1,\dots,m$. We obtain profit w_j if the item is packed into all bins (i.e., packed into the common assortment) and profit v_{jk} if item j is packed into bin k, but there is at least one bin in which we do not pack it. Using this notation, we can model the problem as the following integer program:

$$
(\mathsf{APP})\quad
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{j=1}^{n} w_j x_j + \sum_{j=1}^{n}\sum_{k=1}^{m} v_{jk} y_{jk} & \\
\textbf{subject to} & \displaystyle\sum_{j=1}^{n} (x_j + y_{jk}) \leq K & \forall k \in \{1,\dots,m\} \\
& x_j + y_{jk} \quad \leq 1 & \forall j \in \{1,\dots,n\}, k \in \{1,\dots,m\} \\
& x_j, y_{jk} \quad \in \{0,1\} & \forall j \in \{1,\dots,n\}, k \in \{1,\dots,m\}
\end{array}
$$

where

$$
x_j = \begin{cases} 1, & \text{if item } j \text{ is packed into the common assortment} \\ 0, & \text{else} \end{cases}
$$

and

$$
y_{jk} = \begin{cases} 1, & \text{if item } j \text{ is packed into a bin } k \text{ (but not into the common assortment)} \\ 0, & \text{else.} \end{cases}
$$

The profits w_j and v_{jk} are composed of estimated revenue and cost as follows: Let r_{jk} be the estimated revenue of product j in store k, c_{djk} the cost of type d (e.g., procurement, transportation, storage cost) of product j in store k, c^x the assignment cost for assigning a product to the common assortment, and c^y_k the assignment costs for assigning a product to the local assortment of store k. Then, we can write the objective function of the APP as

$$
\sum_j \sum_k r_{jk}(x_j + y_{jk}) - \sum_j \sum_k \sum_d c_{djk}(x_j + y_{jk}) - \sum_j \sum_k (c^x x_j + c^y_k y_{jk}).
$$

where

$$
w_j = \sum_k \left(r_{jk} - \sum_d c_{djk} - c^x \right) \quad \text{and} \quad v_{jk} = r_{jk} - \sum_d c_{djk} - c^y_k.
$$

Thus, the estimation of the parameters w_j and v_{jk} is essentially the estimation of the revenue and costs of the products.

The main goal of this model for the APP is to decide which products to place in the common assortment and which in the local assortments, given estimated parameters w_j and v_{jk}.

3 Computational Complexity of the APP

In this section, we analyze the computational complexity of the APP. Theorem 2 states that the problem is strongly NP-hard. Thus, unless P = NP, there is no algorithm that solves the APP exactly in polynomial time.

Theorem 2. *The APP is strongly NP-hard.*

Proof. We use a reduction from the satisfiability problem SAT. An instance of SAT consists of n boolean variables $x_1,\dots,x_n$ and m clauses $C_1,\dots,C_m$ of arbitrary length.

Given an instance of SAT, we construct an instance of APP with $2n+1$ items and $n+m+1$ bins of size $n+1$ as follows: There is an item for each positive literal denoted by x_i, an item for each negative literal denoted by $\bar{x}_i$, and one additional item z. The first n bins $A_1,\dots,A_n$ correspond to the variables and the next m bins $B_1,\dots,B_m$ correspond to the clauses. Additionally, there is one extra bin Z. We define the following profits for the items:

$$v_{x_i,A_k} := \begin{cases} 1, & \text{if } k=i \\ 0, & \text{else} \end{cases} \qquad v_{x_i,B_k} := \begin{cases} 1, & \text{if } \bar{x}_i \in C_k \\ 0, & \text{else} \end{cases} \qquad v_{x_i,Z} := 0$$

$$v_{\bar{x}_i,A_k} := \begin{cases} 1, & \text{if } k=i \\ 0, & \text{else} \end{cases} \qquad v_{\bar{x}_i,B_k} := \begin{cases} 1, & \text{if } x_i \in C_k \\ 0, & \text{else} \end{cases} \qquad v_{\bar{x}_i,Z} := 0$$

$$v_{z,A_k} := 0 \quad \forall k \qquad v_{z,B_k} := 0 \quad \forall k \qquad v_{z,Z} := N$$

For the items corresponding to the literals, we set $w_{x_i} := M$ and $w_{\bar{x}_i} := M$, where $M > n+m+1$ is a large integer. For the additional item z, we set $w_z := N$, where $N > M$ is an even larger integer.

We now show that the given instance of SAT admits a satisfying assignment if and only if there exists a packing with objective value at least $nM+n+m+N$ for the constructed instance of the APP.

First, suppose that the SAT instance admits a satisfying assignment. Then, we pack item x_i ($\bar{x}_i$) into the common assortment if variable x_i from the SAT instance is set to TRUE (FALSE) in this assignment. This results in a total profit of nM from the common assortment. Now, exactly one additional item fits into each bin. We put item z into bin Z and, thus, obtain an additional profit of N. For each i, exactly one of the two items $x_i, \bar{x}_i$ is packed into the common assortment and, thus, we can put the other one into bin A_i, which yields an additional profit of n when considering all n bins A_i. Moreover, we know that each clause C_j is satisfied, so there exists a literal $x_i \in C_j$ ($\bar{x}_i \in C_j$) for which the corresponding variable x_i is set to TRUE (FALSE). By the construction above, this means that item x_i ($\bar{x}_i$) is packed into the common assortment and $\bar{x}_i$ (x_i) is not. Thus, we can pack item $\bar{x}_i$ (x_i) into bin B_j and obtain an additional profit of 1. Hence, since all clauses are satisfied, we obtain an additional profit of m in this way, which yields a total profit of $nM+n+m+N$.

Now suppose that there exists a packing with objective value at least $nM+n+m+N$ for the constructed instance of APP. In this packing, item z cannot be in the common assortment since, otherwise, the most profitable way to pack items is to put n items into the common assortment (as $M > n+m+1$), which results in an overall profit of $N+nM$. Moreover we can assume without loss of generality that item z is packed into the last bin Z since this yields a profit of N and the alternatives of packing another item into bin Z or packing another item into the common assortment yield profits 0 and $M < N$, respectively. Also, since $M > m+n+1$, we have to put n items into the common assortment in order to obtain profit nM. In order to obtain the remaining profit of $n+m$, each items packed into one of the remaining $n+m$ empty slots of the bins A_j and B_j has to yield a profit of one. Thus, for each i, one of the items x_i and $\bar{x}_i$ must *not* be packed into the common assortment since, otherwise, no item can generate further profit in bin A_i. Hence, we can define a truth assignment of the variables in the SAT instance by setting variable x_i to TRUE (FALSE) if item x_i ($\bar{x}_i$) is packed into the common assortment. This truth assignment satisfies all clauses since an item x_i ($\bar{x}_i$) that generates a profit of 1 is packed into each bin B_j, which means that the literal $\bar{x}_i \in C_j$ ($x_i \in C_j$) satisfies the clause.

□

4 Algorithmic Approaches

As it turns out (see Section 5), commercial solvers are unable to obtain optimal solutions within a reasonable amount of time even for instances with 300 stores and 15.000 products. This motivates the development of algorithms that run fast and produce close to optimal solutions.

4.1 2-Approximation Algorithm

The proof of Theorem 2 did not provide us with any gap that would prevent us from deriving approximation algorithms. Indeed, a 2-approximation algorithm is easy to obtain. For a given instance of the APP, let W denote the optimum objective value of the corresponding instance with objective function $\sum_j w_j x_j$ and let V denote the optimum objective value of the corresponding instance with objective function $\sum_{jk} v_{jk} y_{jk}$. These values can be computed easily in polynomial time. From now on, we refer to such a pair of corresponding optimal solutions as *restricted solutions*. Letting OPT denote the optimum objective value of the original APP instance, we then have

$$\mathrm{OPT} \leq W + V \leq 2 \cdot \max\{W, V\}$$

Therefore, choosing the better one of the two restricted solutions yields a 2-approximation. With a more careful analysis, one can show the following lemma:

Lemma 1. *We have* $\mathrm{OPT} \leq \left(1 + \min\left\{\frac{V}{W}, \frac{W}{V}\right\}\right) \cdot \max\{W, V\}$.

Proof. We have

$$\mathrm{OPT} \leq V + W = W \cdot \frac{V}{W} + W = \left(1 + \frac{V}{W}\right) \cdot W \quad \text{and}$$
$$\mathrm{OPT} \leq V + W = V + V \cdot \frac{W}{V} = \left(1 + \frac{W}{V}\right) \cdot V. \quad \square$$

The previous lemma shows that the worst case approximation factor of 2 can only occur if $W = V$. Indeed, the approximation factor of the algorithm that chooses the better of the two restricted solutions is heavily dependent on the distribution of the values w_j and v_{jk}. In particular, if w_j and v_{jk} are contained in some small interval for all j, k, the approximation ratio is small:

Lemma 2. *Suppose that there exist* ω *and* $0 \leq \varepsilon \leq 1$ *such that*

$$(1-\varepsilon) \cdot \omega \leq w_j \leq \omega \qquad \forall j \quad \textit{and}$$
$$(1-\varepsilon) \cdot \omega \leq m \cdot v_{jk} \leq \omega \qquad \forall j, k.$$

Then, $\mathrm{OPT} \leq \frac{1}{1-\varepsilon} \cdot \max\{W, V\}$.

Proof. Assume that there are L items in the common assortment of a fixed optimum solution. We define W^L as the sum of the L largest values w_j, and, for each k, we define V_k^{K-L} as the sum of the $(K-L)$ largest values v_{jk}. Then, we obtain

$$\mathrm{OPT} \leq W^L + \sum_{k=1}^{m} V_k^{K-L}.$$

However, it also holds that

$$W \geq W^L + (K-L) \cdot (1-\varepsilon) \cdot \omega \quad \text{and}$$
$$V \geq \sum_{k=1}^{m} V_k^{K-L} + L \cdot (1-\varepsilon) \cdot \omega.$$

In particular, if $W \geq V$,

$$\frac{\mathrm{OPT}}{\max\{W,V\}} \leq \frac{W^L + \sum_{k=1}^{m} V_k^{K-L}}{W^L + (K-L)(1-\varepsilon)\omega} \leq \frac{W^L + (K-L)\omega}{W^L + (K-L)(1-\varepsilon)\omega} \leq \frac{1}{1-\varepsilon}.$$

If $V \geq W$, we obtain an analogous inequality.

□

The main difficulty of the APP is to choose the items that belong to the common assortment as shown with the following lemmas.

Lemma 3. *If the set of items in the common assortment of an arbitrary optimum solution is given and has cardinality L, an optimum solution can be obtained by packing the $K-L$ items with the largest values v_{jk} into each bin k.*

Proof. This is immediate due to the structure of the problem: the bins are now independent and all items have the same size.

□

Lemma 4. *If K is fixed independently of the input, the APP can be solved in polynomial time.*

Proof. If K is fixed independently of the input, we can simply enumerate all the $\sum_{i=0}^{K}\binom{n}{i} \in \mathscr{O}(n^K)$ possible sets of items in the common assortment and, for each such set, compute the solution obtained by filling the bins as in Lemma 3. The best one among these solutions must be optimal.

□

4.2 Greedy Heuristic

We now present a greedy heuristic that is used in Section 5 to solve large-scale instances of the APP. The idea of the algorithm is to first neglect the advantages of using the common assortment and start with the best local packing for each store (independent of the others). Then, in each step, the algorithm adds the item that currently grants the largest gain in total profit to the common assortment. Denoting the current common assortment by C and the current set of items in bin k by I_k, the algorithm can be formulated as follows:

Algorithm 1 Algorithm for the assortment planning problem

1: Let $C = \emptyset$ and, for each bin k, let I_k be the set containing the K items j with the highest values v_{jk}.
2: For $j \notin C$, let $u_j = w_j - \left(\sum_{k:\, j\notin I_k} \min_{l\in I_k} v_{lk}\right) - \sum_{k:\, j\in I_k} v_{jk}$.
3: If $u_j \leq 0$ for all j or $|C| = K$, stop; else add the item $j \notin C$ with the largest value u_j to the common assortment C, update the sets I_k by removing j from I_k if it is contained in I_k and removing an item j' with minimum value $v_{j'k}$ from I_k otherwise, and go to step 2.

Each time we update I_k, its size is reduced by one since we remove one item and put it into C. Observe that Algorithm 1 also works in the case where the capacities of the bins differ. Then, in the first step, for each bin k, we pack the best K_k items, where K_k denotes the individual capacity of bin k. Moreover, the second stopping criterion changes to $|C| = \min_k K_k$. The running time of Algorithm 1 is in $\mathscr{O}(nm(\log(n)+K))$.

As shown at the beginning of this section, it can easily be seen that choosing the better one of the best packings with $|C| = K$ (ALG_2) and $C = \emptyset$ (ALG_3) yields an approximation ratio of 2. If we modify Algorithm 1 (ALG_1) slightly so that it compares the computed solution with the one obtained by ALG_2 and chooses the better one, it also obtains this approximation guarantee.

Additionally, the following preprocessing strategy can be used to identify items that will never be contained in the common assortment:

Preprocessing rule

Let the restricted packing computed by ALG_3 be given. Let V_k denote the set of items in bin k in this packing. Then, for each item j, in order to be eligible to be packed into the common assortment, its *bonus* $b_j := w_j - \sum_{k=1}^{m} v_{jk}$ must exceed the value $\sum_{k=1}^{m} \max\{0, \min_{l \in V_k} v_{lk} - v_{jk}\}$. We define the *residual bonus* of item j as $r_j := b_j - \sum_{k=1}^{m} \max\{0, \min_{l \in V_k} v_{lk} - v_{jk}\}$. Then, we can establish the following preprocessing rule: If $r_j < 0$, then item j is not in the common assortment in any optimal packing, i.e, we may set $x_j := 0$ (cf. Section 2).

5 Experimental Results

In this section, we present computational experiments in order to compare the solution quality obtained by ALG_1, ALG_2, and ALG_3. In particular, we are interested in how large the common assortment profits w_j must be in relation to the local profits v_{jk} in order to see substantial benefits from mixing common and local assortments as in ALG_1 (when compared to using only the common assortment as in ALG_2 or only the local assortments as in ALG_3).

We randomly generate the values v_{jk} and w_j. However, it seems to be a reasonable assumption that the values v_{jk} are dependent for a fixed item j although the cost of providing item j might vary for different stores (e.g., transportation cost). Therefore, we consider three scenarios where we draw values v_j uniformly at random from $[0,1]$ independently for all j and then

- set $v_{jk} := v_j$ for all k (*total dependence*), or
- draw r_k uniformly from $[-0.5p, 0.5p]$ and set $v_{jk} := \max(0, v_j + r_k)$, where p is a model parameter (*intermediate dependence*), or
- draw all values v_{jk} uniformly and independently from $[0,1]$ (*total independence*).

Observe that, for $p = 0$, the first and the second scenario are identical. In order to generate the values w_j, we draw values q_j uniformly at random from $[0.95, 1.05]$ and set $w_j := q_j b \sum_k v_{jk}$, where b represents the financial gains when a product is in the common assortment (e.g., from economies of scale or recognition value). We consider 100 equidistant values of b in $[1,2]$ and generate 100 instances for each of these values and four different settings concerning the dependence of the values v_{jk} (total independence, intermediate dependence with $p = 0.75$ and $p = 0.95$, and total independence). The optimal profit for each instance is calculated by solving the IP formulation given in Section 2 using Gurobi 6.5.

We observe that, when b is too small or too large, algorithms ALG_2 and/or ALG_3 already yield close to optimal solutions (cf. Figure 1). If a company estimates the value b in such a way, the assortment planning that we propose might not be necessary. Therefore, for each of the four settings concerning dependence of the values v_{jk}, we concentrate on three values of b that are neither too small nor too large. In the setting of total independence, we consider $b \in \{1.2, 1.35, 1.5\}$, for $p = 0.75$ and total dependence, we consider $b \in \{1.01, 1.05, 1.09\}$, and for $p = 0.95$, we consider $b \in \{1.04, 1.09, 1.14\}$. Moreover, we consider two different instance sizes (small and large), where $(n,m,K) = (1500, 50, 750)$ and $(n,m,K) = (50.000, 150, 25.000)$, respectively. For all considered instances, ALG_1 obtains nearly optimal solutions (i.e., the average ratio $\frac{\text{OPT}}{\text{ALG}_1}$ of the profits of an optimal solution and the algorithm is below 1.01). Table 1 shows the average ratios $\frac{\text{OPT}}{\text{ALG}_i}$ for $i = 2,3$ obtained by ALG_2 and ALG_3. Since the large instances could not be solved to optimality within a reasonable amount of time by using Gurobi, we provide the ratios $\frac{\text{ALG}_1}{\text{ALG}_i}$ instead for these instances. Here, we observe that the gain in profit from using an optimized assortment compared to one of the solutions produced by ALG_2 or ALG_3 can be up to 20%. In Figure 2, we compare the running times of Gurobi and Algorithm 1. It can be seen that, even for instances with $(n,m,K) = (15.000, 300, 7500)$, Gurobi already needs 30 hours in order to solve a single instance, whereas Algorithm 1 finishes in less than ten minutes.

From a practical point of view, a retailer should estimate the degree of independence between items as well as the influences of economies of scale (i.e., estimate the values v_{jk} and w_j). Then she can decide if the benefits of using an optimized assortment planning strategy justify a change in her assortment planning. Observe that the benefit of using an optimized assortment planning strategy seems to be independent of the size of the instance.

Table 1 Average ratios $\frac{\text{OPT}}{\text{ALG}_i}$ for $i = 2,3$ (for small instances) and $\frac{\text{ALG}_1}{\text{ALG}_i}$ (for large instances).

Value b: / **Instance Size:** / **Dependence:**	small / small	medium / small	large / small	small / large	medium / large	large / large
total dependence	1.01|1.02	1.00|1.05	1.00|1.09	1.01|1.02	1.00|1.05	1.00|1.09
$p = 0.75$	1.06|1.01	1.04|1.03	1.03|1.06	1.06|1.01	1.04|1.03	1.03|1.06
$p = 0.95$	1.07|1.02	1.05|1.05	1.04|1.08	1.08|1.02	1.05|1.05	1.04|1.08
total independence	1.18|1.01	1.09|1.05	1.04|1.11	1.20|1.01	1.10|1.03	1.05|1.09

Fig. 1 On the x-axis, the value b is shown. On the y-axis, the ratio $\frac{\text{OPT}}{\text{ALG}_i}$ is shown for $i = 2$ (stars) and $i = 3$ (crosses). We observe that $\min(\text{ALG}_2, \text{ALG}_3)$ is maximum when $b \approx 1.35$ (in the case of total independence).

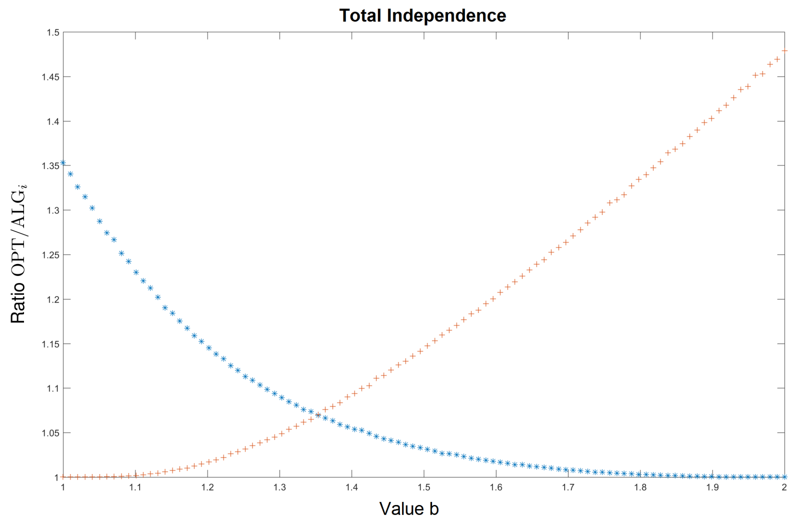
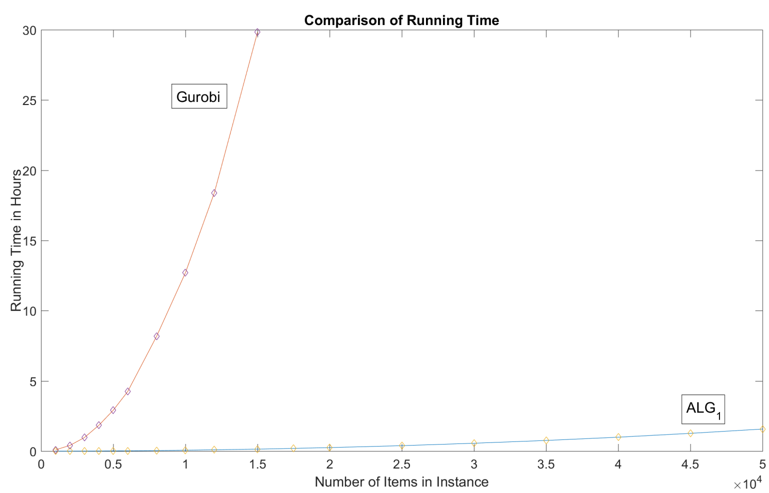


Fig. 2 Comparison of the running time of Gurobi and Algorithm 1. We consider instances with $m = 300$ and $K = \frac{n}{2}$, where n is given on the x-axis. The running time is given in hours on the y-axis. An estimation of the the running time of our algorithm also leads to 14 hours ($n = 150.000$), 26 days ($n = 10^6$), and 72.000 years ($n = 10^9$) on a standard desktop computer.



Next, we consider mixed scenarios that are represented by a (4×3)-matrix, where each row represents one of the four scenarios we use to generate the values v_{jk} (total dependence, $p = 0.75$, $p = 0.95$, total independence), and each column corresponds to a choice of the value b (small, medium, large). This provides us with twelve categories and each entry of the matrix indicates the fraction of items that are drawn from this category. Table 2 shows the results.

Table 2 Ratios $\frac{\text{OPT}}{\text{ALG}_i}$ for different mixed scenarios. Again, we consider small and large instances.

Scenario	**Description**	$\frac{\text{OPT}}{\text{ALG}_2} \mid \frac{\text{OPT}}{\text{ALG}_3}$	$\frac{\text{ALG}_1}{\text{ALG}_2} \mid \frac{\text{ALG}_1}{\text{ALG}_3}$
$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.25 & 0.25 \\ 0 & 0.25 & 0.25 \end{pmatrix}$	Item profits are rather independent between the stores and economies of scale apply well.	1.02|1.24	1.02|1.23
$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.33 & 0.33 & 0.33 \end{pmatrix}$	Item profits are independent between the stores and, for some items, economies of scale apply.	1.04|1.11	1.05|1.09
$\begin{pmatrix} 0 & 0 & 0 \\ 0.33 & 0 & 0 \\ 0.33 & 0 & 0 \\ 0.33 & 0 & 0 \end{pmatrix}$	Item profits are rather dependent between the stores and economies of scale do not apply.	1.07|1.09	1.07|1.09
$\begin{pmatrix} 0 & 0 & 0 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \end{pmatrix}$	There is a mixture of dependent and independent item profits and, for some items, economies of scale apply.	1.01|1.30	1.02|1.29

6 Conclusion

We have formulated a stylized model for the regionalized assortment planning problem (APP) and have shown that solving the APP can lead to significant profit gains for a retailer with multiple chain stores. However, it seems computationally impossible to solve its natural integer programming formulation for large-scale instances. Therefore, we proposed a local improvement heuristic that computes close to optimal solutions in polynomial time. In a next step, we will test this algorithm with real world data. For future research, we propose extensions of the model such as sub-regions (here, we also obtain a profit gain when a product is placed in a certain set of stores) or individual item sizes.

References

1. Borin, N., Farris, P.W.: A sensitivity analysis of retailer shelf management models. J. Retail. **71**, pp. 153-171 (1995)
2. Borin, N., Farris, P.W., Freeland, R.: A model for determining retail product category assortment and shelf space allocation. Decis. Sci. **25**, pp. 359-384 (1994)
3. Chen, M.-C., Lin, C.-P.: A data mining approach to product assortment and shelf space allocation. Expert Syst. Appl. **32**, pp. 976-986 (2007)
4. Corsten, H., Kasper, B.: Entwurf mehrstufiger Sortimentsplanungsmodelle in Handelsunternehmungen. Schr. zum Produktionsmanage. **96**, Kaiserslautern (2016)
5. Farias, V.F., Jagabathula, S., Shah, D.: A nonparametric approach to modeling choice with limited data. Manage. Sci. **59**, pp. 305-322 (2013)
6. Fisher, M., Vaidyanathan, R.: A demand estimation procedure for retail assortment optimization with results from implementations. Manage. Sci. **60**, pp. 2401-2415 (2014)
7. Gaur, V., Honhon, D.: Assortment planning and inventory decisions under a locational choice model. Manage. Sci. **52**, pp. 1528-1543 (2006)
8. Grewal, D., Levy, M., Mehrotra, A., Sharma, A.: Planning merchandising decisions to account for regional and product assortment differences. J. Retail. **75**, pp. 405-424 (1999)
9. Honhon, D., Gaur, V., Seshadri, S.: Assortment planning and inventory decisions under stockout-based substitution. Oper. Res. **58**, pp. 1364-1379 (2010)
10. Hübner, A.H., Kuhn, H.: Retail category management: State-of-the-art review of quantitative research and software applications in assortment and shelf space management. Omega - Int. J. Manage. Sci. **40**, pp. 199-209 (2012)
11. Hwang, M., Bronnenberg, B.J., Thomadsen, R.: An empirical analysis of assortment similarities across US supermarkets. Market. Sci. **29**, pp. 858-879 (2010)
12. Li, G., Rusmevichientong, P.: A greedy algorithm for the two-level nested logit model. Oper. Res. Lett. **42**, pp. 319-324 (2014)
13. Kök, A.G., Fisher, M.L.: Demand estimation and assortment optimization under substitution: methodology and application. Oper. Res. **55**, pp. 1001-1021 (2007)
14. Kök, A.G., Fisher, M.L., Vaidyanathan, R.: Assortment planning: Review of literature and industry practice. In: Agrawal, N., Smith, S.A. (eds) Retail Supply Chain Manag. **223** pp. 99-153. Springer US (2009)
15. Mantrala, M.K., Levy, M., Kahn, B.E., Fox, E.J., Gaidarev, P., Dankworth B., Shah, D.: Why is assortment planning so difficult for retailers? A framework and research agenda. J. Retail. **85**, pp. 71-83 (2009)
16. Rajaram, K.: Assortment planning in fashion retailing: methodology, application an analysis. Eur. J. Oper. Res. **129**, pp. 186-208 (2001)
17. Rooderkerk, R.P., van Heerde, H.J.: Robust optimization of the 0-1 knapsack problem: Balancing risk and return in assortment optimization. Eur. J. Oper. Res. **250**, pp. 842-854 (2016)
18. Rooderkerk, R.P., van Heerde, H.J., Bijmolt, T.H.A.: Optimizing retail assortments. Market. Sci. **32**, pp. 699-715 (2013)
19. Rusmevichientong, A., Shmoys, D., Tong, C., Topaloglu, H.: Assortment optimization under the multinomial logit model with random choice parameters. Prod. & Oper. Manage. **23**, pp. 2023-2039 (2014)
20. Shin, H., Park, S., Lee, E., Benton, W.C.: A classification of the literature on the planning of substitutable products. Eur. J. Oper. Res. **246**, pp. 686-699 (2015)
21. Syring, D.: Bestimmung effizienter Sortimente in der operativen Sortimentsplanung. Berlin (2004)
22. Talluri, K., van Ryzin, G.: Revenue management under a general discrete choice model of consumer behavior. Manage. Sci. **50**, pp. 15-33 (2004)
23. Urban, T.L.: An inventory-theoretic approach to product assortment and shelf-space allocation. J. Retail. **74**, pp. 15-35 (1998)
24. Vulcano, G., van Ryzin, G., Ratliff, R.: Estimating primary demand for substitutable products from sales transaction data. Oper. Res. **60**, pp. 313-334 (2012)
25. Wang, R.: Capacitated assortment and price optimization under the multinomial logit model. Oper. Res. Lett. **40**, pp. 492-497 (2012)