*Optimization Methods and Software*
Vol. 00, No. 00, January 2008, 1–12

**RESEARCH ARTICLE**

# On the lot-type design problem

Constantin Gaul, Sascha Kurz* and Jörg Rambau

*Department of Mathematics, Physics, and Computer Science, University of Bayreuth, 95440 Bayreuth, Germany*

(*Received 00 Month 200x; in final form 00 Month 200x*)

We consider the problem of approximating the branch and size dependent demand of a fashion discounter with many branches by a distributing process being based on the branch delivery restricted to integral multiples of lots from a small set of available lot-types. We propose a formalized model which arises from a practical cooperation with an industry partner. Besides an integer linear programming formulation we provide an appropriate primal heuristic for this problem.

**Keywords:** $p$-median problem;facility location problem;integer linear program formulation;primal heuristic;real world data;location-allocation;

**AMS Subject Classification**: 90B80*; 90C59; 90C10;

## 1. Introduction

Fashion discounters usually achieve only small profit margins. Their economic success depends mostly in the ability to meet the customers' demands for individual products. More specifically: offer exactly what you can sell to your customers. This task has two aspects: offer what the customers would like to wear (attractive products) and offer the right volumes in the right places and the right sizes (demand consistent branch and size distribution).

In this paper we deal with the second aspect only: meet the branch and size specific demand for products as closely as possible. Our industry partner is a fashion discounter with more than 1 000 branches most of whose products are never replenished, except for the very few "never-out-of-stock"-products (NOS products): because of lead times of around three months, apparel replenishments would be too late anyway. In most cases the supplied items per product and apparel size lie in the range between 1 and 6. Clearly, it is a challenge to determine a good estimate for the branch and size dependent demand, but besides a few practical comments on this problem we will blind out this aspect of the problem completely. We assume that for a product we know the mean demand for each size in each branch. In general, this will be a fractional value for each branch-size combination.

The problem studied in this article is motivated by a special feature of the ordering process: For each product that hits the shelves, the internal stock-turnover has to distribute around 10 000 pieces among the around 1 000 branches, correctly assorted by size. This would mean 10 000 picks with high error probability in the

*Corresponding author. Email: sascha.kurz@uni-bayreuth.de

ISSN: 1055-6788 print/ISSN 1029-4937 online
© 2008 Taylor & Francis
DOI: 10.1080/1055678xxxxxxxxxxxxx
http://www.informaworld.com

central-warehouse (in our case in the high-wage country Germany). In order to reduce the handling costs and the error proneness in the central warehouse, all products are ordered in multiples of so-called *lot-types* from the suppliers who in general are located in extremely low-wage countries.

A lot-type specifies a number of pieces of a product for each available size, e.g., (2,2,2,2,2) if the sizes are (S, M, L, XL, XXL) means two pieces of each size. A *lot* of a certain lot-type is a foiled pre-pack that contains as many pieces of each size as specified in its lot-type. Assume that, e.g., the 10 000 pieces of a product arrive in 1 000 lots of lot-type (2,2,2,2,2) and every branch receives exactly one lot of that lot-type, then the number of error-prone picks in the central warehouse has been reduced to $\frac{1}{10}$.

The real-world is, of course, more complicated than this example. This is mainly because the mean demands for each branch and each size differ. Moreover, it is not possible to order the supply in 1 000 lot-types, one for each branch. The suppliers accept at most five or so different lot-types for an order. This leads to the problem that even if we know the branch and size dependent mean demand we cannot simply supply the branches by volumes for each size according to the demands rounded to the nearest integer. We have to choose lot-types first and specify the order volume in terms of multiplicities of these lot-types, which maybe much more restrictive. Similarly, the volume delivered to the branches must be specified in terms of multiplicities of the ordered lot-types.

So we face an approximation problem: which (integral) multiples of which (integral) lot-types should be supplied to a branch in order to meet a (fractional) mean demand as closely as possible? We call this specific demand approximation problem the *lot-type design problem (LDP).*

## 1.1. *Related Work*

The LDP is closely related to the extensively studied $p$-median and the facility location problem: a set of customers has to be connected to exactly $p$ locations, where in the capacitated version each customer has a demand and each location can only serve a bounded total demand (see [8, 10] for recent progress in heuristic approaches and [1] for recent computational results on large instances of the uncapacitated $p$-median problem; see [3, 4] for recent results on ILP techniques to the capacitated $p$-median problem). The $p$-median problem models (beyond the true location problem), e.g., clustering problems in various contexts. Loads of heuristics and ILP techniques have been applied to it. Nevertheless the first constant-factor approximation algorithm, based on LP rounding, was given not until 1999 by Charikar, Guha, Tardos, and Shmoys [5]. We can not take advantage of this result because our ILP model solves nearly as fast as its LP relaxation, the basis for the approximation algorithm.

Although the name "lot" suggests a similarity to the classical lot sizing problems (see [9, 12] for a classification, references, and mixed integer programming formulations) there is not much that the LDP has in common with lot sizing. The main difference is that in the LDP the order volumes for all branches and sizes are not independent because they must fit to the lot-type patterns that are in turn consequences of our lot-type selection decision. Moreover, in the LDP we are only concerned with a single period problem (no replenishment). It is conceivable that some sort of multi-period LDP is relevant in other applications, though.

There is a line of research in retail revenue management that links the classical revenue management decision in fashion retail (mark-downs) to the inventory decisions (see [6] for a survey). We did not find any published research that considered combinatorial restrictions (like the one implied by the lot-type based supply) for the start inventory decisions. Moreover, in our case, the inventory restrictions we face are the result of our lot-type design decisions, and such a two-stage inventory optimization we could not find anywhere in the literature.

Another interesting connection can be drawn to *assortment optimization* (see [7] for recent progress). In this problem, a start inventory for a portfolio of styles of a piece of garment is computed so as to maximize expected revenue. The practical difficulty of this problem lies in the estimation of the demand depending on the stock levels, since substitution effects (the customer may buy the blue T-shirt instead of the sold-out black one) contaminate the historical data. The difficulties tackled in assortment optimization are much more due to the stochasticity of the problem than on combinatorial restrictions on the shapes of assortments, which have, to the best of our knowledge, not been investigated so far.

There are traces of so-called *pre-pack optimization work-flows* of commercial providers of revenue management software. It is, however, not easy to find out what the exact problem is that they solve. There is, e.g., a sketch of a solution approach in a white paper of Chettri and Sharma from Cognizant Technology Solutions.[1] A closer inspection shows that in this paper the pre-pack based supply can be completed by single items so that exactly our approximation problem is by-passed (at the cost of a more complicated handling).

We note that all of the above connections motivate extended versions of the cited problems by simply adding our lot-type design phase and the corresponding start inventory restrictions to the problem. Only the $p$-median problem captures a design phase at all (which lot-types to choose in the first place); it neglects the choice of multiplicities and the total capacity delivered, though.

Instead of adding the lot-type design phase to one of the more retail oriented problem classes like assortment optimization (which makes perfect sense because also in our case there exist difficulties in the demand estimation), we chose to investigate the pure, deterministic lot-type design problem first (which generalizes the $p$-median problem) in order to start with the simplest possible new problem class modeling a crucial aspect of the work-flow under consideration.

## 1.2. *Our contribution*

We define the lot-type design problem LDP for the order and distribution process of a fashion discounter with many branches. This optimization problem has, to the best of our knowledge, not been studied in the literature so far. We briefly relate the LDP to the $p$-median problem and show that the LDP is strictly more general. We present an integer linear programming model for the LDP that has tight LP relaxations on our instances. Commercial ILP solver can therefore solve our large-scale real-world instances in at most half an hour. Moreover, we devise a fast primal any-time heuristic SFA that achieves optimality gaps of around 1 % in a second on real-world instances of our industry partner. SFA can therefore be used interactively in practice in the process of order negotiations. Meanwhile, a prototype of SFA was put to operation in our industry partner's IT.

## 1.3. *Outline of the paper*

In Section 2, we give the formal statement of the lot-type design problem LDP and relate it to the $p$-median problem. In Section 3, we provide an integer linear

[1] `http://www.cognizant.com/html/content/bluepapers/Pre_Pack_Optimization.pdf`

programming model for the LDP. In Section 4 we present a primal heuristic for the LDP, which we applied to our real world problem. We give some real-world numerical data on the optimality gap of our heuristic, before we draw some conclusions and possible future directions in Section 5.

## 2. Formal problem statement

Consider a fashion discounter with branches $b \in \mathcal{B}$ who wants to place an order for a certain product that can be obtained in sizes $s \in \mathcal{S}$ and that can be pre-packed in lot-types $l \in \mathcal{L}$. Each lot-type is a vector $(l_s)_{s\in\mathcal{S}}$ specifying the number of pieces of each size contained in the pre-pack. Only $k$ different lot-types from $\mathcal{L}$ are allowed in this order, and each branch receives only lots of a single lot-type. We are given lower and upper bounds $\underline{I}, \overline{I}$ for the total supply of this product. Moreover, we assume that a the branch and size dependent mean demand $d_{b,s}$ for the corresponding type of product is known to us.

The original goal is to find a set of at most $k$ lot-types, an order volume for each of these chosen lot-types, and a distribution of lots to branches such that the revenue is maximized. In order to separate the order process from the sales process (which involves mark-downs, promotions, etc.), we restrict ourselves in this paper to the minimization of the distance between supply and mean demand defined by a vector norm.

Let us first state the assumptions under which it is plausible that a solution to our optimization problem defined below will yield a good revenue.

*Assumption 2.1*: We make the following assumptions:

(1) All products can be considered separately
(2) There is a significant negative correlation between distance of supply from mean demand and revenue (i.e., on average we have: the smaller the distance, the larger the revenue)

Moreover, in order to simplify the presentation, we assume that each branch must be supplied, i.e., has to receive a strictly positive number of lots of some lot-type.

We claim that, under these assumptions, the order and distribution according to any (almost) optimal solution of the following *lot-type design problem (LDP)* will have a significant desirable impact on the revenue:

*Problem 2.2*: The *Lot-Type Design Problem (LDP)* is the following optimization problem:

*Instance:* We are given

- a set of branches $b \in \mathcal{B}$
- a set of sizes $s \in \mathcal{S}$
- a mean demand table $d_{b,s}$, $b \in \mathcal{B}$, $s \in \mathcal{S}$
- a norm $\|\cdot\|$ on $\mathbb{R}^{\mathcal{B}\times\mathcal{S}}$
- a set $\mathcal{L}$ of feasible lot types $(l_s)_{s\in\mathcal{S}} \in \mathbb{N}_0^{\mathcal{S}}$
- a maximal number $M \in \mathbb{N}$ of possible multiplicities
- a maximal number $k \in \mathbb{N}$ of lot types to use
- lower and upper bounds $\underline{I}$, $\overline{I}$ for the total supply

*Task:* For each branch $b \in \mathcal{B}$ choose a lot type $l(b) \in \mathcal{L}$ and a number $m(b) \in \mathbb{N}$, $1 \leq m(b) \leq M$ of lots to order for $b$ such that

- the total number of ordered lot types is at most $k$
- the total number of ordered pieces is in $[\underline{I}, \overline{I}]$ (the *total capacity constraint*)
- the distance of the order from the demand measured by $\|\cdot\|$ is minimal

We can easily see that our assumption that each branch must be supplied does not mean any loss of generality: if a branch can remain unsupplied then simply allow zero as a multiplicity for it.

Depending on the norm we pick, we can influence the properties of optimal solutions w.r.t. fairness among branches and sizes and robustness w.r.t. outliers of the mean demand estimations.

| | |
|---|---|
| *$L^\infty$-Norm:* | *Fair* w.r.t. inconsistent supply of *individual branches* in an optimal solution but *not robust* w.r.t. *outliers* in the demand data |
| *$L^1$-Norm:* | *Unfair* w.r.t. inconsistent supply of *individual branches* in an optimal solution but *robust* w.r.t. *outliers* in the demand data |
| *$L^2$-Norm:* | Somewhere inbetween |

For the rest of the paper we deal with the $L^1$-norm for two reasons: first, the mean demand estimations are very delicate, and outliers for some branch-size combinations can hardly be avoided, and second, our industry partner is risk neutral w.r.t. individual branches, i.e., the total revenue is what counts, even when this means that a single branch-size combination individually produces a low revenue.

**Remark 1 :** When we fix the total quantities for all lot-types (at most $k$ non-zero), then the resulting problem models the search for the best *distribution of a given inventory* from the central warehouse among the branches without breaking lots in pieces. This problem is also relevant for our industry partner, since more often than not the actual delivery of the supplier deviates from the order (and only for this order we have a good distribution from the solution of the LDP prior to the order). The resulting inventory then has to be distributed in the best possible way.

**Remark 2 :** Why are we using a total capacity interval and not a fixed total capacity? The first reason lies in number theoretic problems: if, e.g., the total capacity is prime and there is only one lot-type containing more than one piece, then the problem is infeasible. A second reason is that the total capacity to be ordered is (at least partially) based on rough estimations that might contain errors anyway. The third reason comes from the fact that the delivery can deviate up to 5 % from the order volume in practice. An alternative approach would be to turn the total capacity constraint into a soft constraint. However, then we are left with the problem of finding good penalty values for its violation.

**Remark 3:** If we drop the total capacity constraint in the special case $M = 1$, then we obtain a standard model for the well-studied $p$-median problem. This means, in particular, that the LDP is NP-hard. Conversely, even if $M > 1$, the LDP without the total capacity constraint can be reduced to a $p$-median instance: Define for each branch $b$ and each lot-type $l$ the *locally optimal multiplicity* $m^*_{b,l}$ to be the optimizer of $\min_m \sum_{s \in \mathcal{S}} c_{b,l,m}$. It is easy to see that if Lot-type $l$ is chosen for Branch $b$, then it can be chosen w.l.o.g. with the locally optimal multiplicity $m^*_{b,l}$. It remains to choose a lot-type for each branch, which can be solved by a $p$-median computation with distance $c_{b,l,m^*_{b,l}}$ between $b$ and $l$.

Unfortunately, in the presence of the total capacity constraint we have found no way to take advantage from the relationship to the $p$-median problem. This is mainly due to the fact that the locally optimal multiplicities might become infeasible. This can easily be seen from the following example.

**Example 2.3** Assume, there is one branch, one size, one lot-type $l = 1$, a demand of 2, and an identical upper and lower bound of the total supply of 1. Then, obviously a supply of two lots would be locally optimal, but would also violate the total capacity constraint.

This seems too far-fetched to be surprising. In particular, the total number of pieces in the previous example was different from the total demand. However: even if the total demand $\sum_{b \in \mathcal{B}} \sum_{s \in \mathcal{S}} d_{b,s}$ is in the center of an arbitrarily large interval $[\underline{I}, \overline{I}]$, the total capacity constraint can lead to the infeasibility of locally optimal multiplicities. This can be seen in the following example.

**Example 2.4** Consider an instance of the LDP with a single size $s$ and a single lot-type $l = l_s = 1$.

Assume, $\underline{I} = I - i$ for some positive integer $i$, $\overline{I} = I + i$ with $I = \sum_{b \in \mathcal{B}} d_{b,s}$. Let $N := |\mathcal{B}|$ be the number of branches, and for a given positive integer $D > 1$ let $d := d_{b,s} = D - \frac{i+j}{N}$ for some positive integer $j$. Whenever $\frac{i+j}{N} < \frac{1}{2}$, we have for each branch that the locally optimal multiplicity is $D$. However, choosing the locally optimal multiplicity throughout leads to $N \cdot D$ pieces. Since $D = d + \frac{i+j}{N}$ and $N \cdot d = I$, we get:

$$N \cdot D = N \left( d + \frac{i+j}{N} \right) = I + i + j = \overline{I} + j \tag{1}$$

Still, the problem is feasible: just deliver $D - 1$ lots to an arbitrary set of $j$ branches and $D$ lots to the rest. By symmetry, this is even optimal.

That means, a consequence of the total capacity constraint is that in an optimal solution there may be an arbitrary subset of branches that is supplied by locally suboptimal multiplicities, and if there is no symmetry as in the example, an optimal solution may be hard to find.

## 3. A mathematical model for the LDP

The problem stated in the previous section can be formulated as an Integer Linear Program, mainly because we restrict ourselves to the $L^1$-norm for measuring the distance between supply and demand.

We use binary variables $x_{b,l,m}$, which are equal to 1 if and only if lot-type $l$ is delivered with multiplicity $m$ to Branch $b$, and binary variables $y_l$, which are 1 if and only if at least one branch in $\mathcal{B}$ is supplied with Lottype $l$. The *deviation* of the demand from the supply if Branch $b$ is supplied by $m$ lots of lot-type $l$ is given by $c_{b,l,m} := \sum_{s\in\mathcal{S}} |d_{b,s} - m \cdot l_s|$.

The following integer linear program models the LDP with $L^1$-norm.

$$\min \quad \sum_{b\in\mathcal{B}} \sum_{l\in\mathcal{L}} \sum_{m=1}^{M} c_{b,l,m} \cdot x_{b,l,m} \tag{2}$$

$$s.t. \quad \sum_{l\in\mathcal{L}} \sum_{m=1}^{M} x_{b,l,m} = 1 \qquad \forall b \in \mathcal{B} \tag{3}$$

$$\sum_{l\in\mathcal{L}} y_l \leq k \tag{4}$$

$$\sum_{m=1}^{M} x_{b,l,m} \leq y_l \qquad \forall b \in \mathcal{B}, \forall l \in \mathcal{L} \tag{5}$$

$$\underline{I} \leq \sum_{b\in\mathcal{B}} \sum_{l\in\mathcal{L}} \sum_{m=1}^{M} \sum_{s\in\mathcal{S}} m \cdot l_s \cdot x_{b,l,m} \leq \overline{I} \tag{6}$$

$$x_{b,l,m} \in \{0,1\} \quad \forall b \in \mathcal{B}, \forall l \in \mathcal{L}, \forall m = 1,\ldots,M \tag{7}$$

$$y_l \in \{0,1\} \quad \forall l \in \mathcal{L} \tag{8}$$

The objective function (2) computes the $L^1$-distance of the supply specified by $x$ from the demand. Condition (3) enforces that each branch is assigned a unique lot-type and a unique multiplicity. Condition (4) models that at most $k$ different lot-types can be chosen. Condition (5) forces the selection of a lot-type whenever it is assigned to some branch with some multiplicity. Finally, Condition (6) ensures that the total number of pieces is in the desired interval – the total capacity constraint.

## 4. The SFA heuristic for the lot-type design problem

Our ILP formulation can be used to solve all real world instances of our business partner in at most 30 minutes by using a standard ILP solver like `CPLEX 11`. Interestingly, the model seems quite tight – most of the time is spent in solving the root LP.

Even without Condition (6) the root LP takes minutes to solve. That is: in practice, even if we found a way to tweak the known LP rounding approximation algorithms for the $p$-median problem like the one in [5], we probably would not achieve a significant speed-up.

Although 30 minutes may mean a feasible computation time for an offline-optimization in many contexts, this is not fast enough for our real world application. The buyers of our retailer need a software tool which can produce a near optimal order recommendation in real time on a standard laptop. For this reason, we present a fast anytime-heuristic, which has only a small gap compared to the optimal solution on a test set of real world data of our business partner.

We explain the idea of the heuristic in the sequel. A very important decision is: which lot-types should be used in the first place? Here one should have in mind that the cardinality $|\mathcal{L}|$ of the set of feasible lot-types is very large compared to the number $k$ of lot-types which can be used for the delivery process. Therefore, in large-scale problems it will be impossible to enumerate all possible combinations unless $\binom{|\mathcal{L}|}{k}$ is small.

In the following, we present the heuristic *Score-Fix-Adjust (SFA)* that

(1) sorts all lot-types according to certain scores, coming from a count for how many branches the lot-type fits best, second best, ... (Score);
(2) fixes $k$-subsets of lot-types in the order of decreasing score sums (Fix);
(3) greedily adjusts the multiplicities so as to achieve feasibility w.r.t. the total capacity constraint (Adjust).

### 4.1. *How to score*

In order to prepare for the selection of lot-types, we define a score for each lot-type. For every branch $b \in \mathcal{B}$ with demand $d_b$ there exists a lot-type $l \in \mathcal{L}$ and a multiplicity $m \in \mathbb{N}$ such that $\|d_b - m \cdot l\|$ is minimal in the set $\{\|d_b - m' \cdot l'\| : l' \in \mathcal{L}, m' \in \{1, \ldots, M\}\}$. So for every branch $b \in \mathcal{B}$ there exists a lot-type that fits best. More general, for a given $k \leq |\mathcal{L}|$ there exist lot-types $l_1, \ldots, l_k$ such that $l_i$ fits $i$th-best if one uses the corresponding locally optimal multiplicity.

Let us examine this situation from the point of view of the different lot-types. A given lot-type $l \in \mathcal{L}$ is the $i$th-best fitting lot-type for a number $\varrho_{l,i}$ of branches in $\mathcal{B}$. Writing these numbers $\varrho_{l,i}$ as a vector $\varrho_l \in \mathbb{N}^k$ we obtain score vectors for all lot-types $l \in \mathcal{L}$.

Now we want to use these score vectors $\varrho_l$ to sort the lot-types of $\mathcal{L}$ in decreasing *approximation quality*. Using the lexicographic ordering $\preceq$ on vectors we can determine a bijective rank function $\lambda : \mathcal{L} \to \{1, \ldots, |\mathcal{L}|\}$: we simply sort the score vectors according to $\preceq$ with ties broken arbitrarily. We extend $\lambda$ to subsets $\mathcal{L}' \subseteq \mathcal{L}$ by $\lambda(\mathcal{L}') = \sum_{l \in \mathcal{L}'} \lambda(l) \in \mathbb{N}$. (An alternative would be the lexicographic total order, but this one worked better for us.)

The main idea behind this scoring method is that by assigning a score to only $|\mathcal{L}|$ many individual lot-types we implicitly get an order on the set of $\binom{|\mathcal{L}|}{k}$ many feasible lot-type combinations. Note that we can easily traverse this order element by element without explicitly generating it beforehand.

### 4.2. *How to fix*

Now, we fix subsets $\mathcal{L}' \subseteq \mathcal{L}$ of cardinality $k$ in decreasing order with respect to $\lambda(\mathcal{L}')$. For each new lot-type combination, we adjust as in Section 4.3 and report every feasible (adjusted) solution that improves the current best.

In principle we could consider all possible selections $\mathcal{L}'$ of $k$ lot-types, but in practice we stop our computations after an adequate time period. The hope is that, by then, we have checked the most promising selections $\mathcal{L}'$ first.

We remark that for large $M$ we can determine an optimal multiplier $m$ for a given branch $b$ and a given lot-type $l$ by binary search.

### 4.3. *How to adjust*

If we generate assignments $l : \mathcal{B} \to \mathcal{L}$, $b \mapsto l(b)$ with corresponding multipliers $m : \mathcal{B} \to \mathbb{N}$, $b \mapsto m(b)$ as in the previous section, then in some cases we will not satisfy the total capacity constraint (6), since it is totally unaccounted by the “Score”-step. Our strategy to satisfy the total capacity constraint (6) is to adjust $m(b)$ afterwards by decreasing or increasing the calculated multipliers, unless Condition (6) is fulfilled by pure chance.

Here we want to use a greedy algorithm and have to distinguish two cases. If $I(l, m)$ is smaller than $\underline{I}$, then we increase $m(b)$ for some $b$, otherwise we have $I(l, m) > \overline{I}$ and we decrease $m(b)$ for some $b$. Our procedure works iteratively, and we assume that the current multipliers are given by $m$. We stop whenever $\underline{I} \leq I(l, m) \leq \overline{I}$ or there are no feasible operations left. Let us describe a single step of the iteration. We restrict our explanation to the case where we want to decrease $m(b)$ for some $b$.

For every branch $b \in \mathcal{B}$ the reduction of $m(b)$ by one produces costs

$$\Delta_b^- = c_{b,l(b),m(b)-1} - c_{b,l(b),m(b)}$$

if the reduction of $m(b)$ by one is feasible (i.e., if $m(b) \geq 1$ if the branch need not be supplied or $m(b) \geq 2$ if the branch must be supplied) and $\Delta_b^- = \infty$ if we do not have the possibility to reduce the multiplier $m(b)$ by one. A suitable data structure for the $\Delta_b^-$ values is a heap, for which the update after an iteration can be done in $O(\log |\mathcal{B}|)$ time. If we reach $I(l, m) < \underline{I}$ at some point, we simply discard this particular lot-type assignment $l$ (because it is alltogether infeasible) and consider the next candidate in the total scoring order.

## 4.4. *Optimality of SFA for $k = 1$*

Since in the case $k = 1$ we can very often loop over all feasible lot-types, it is interesting that in this case SFA always yields an optimal solution (for any norm).

**Lemma 4.1:** *For $k = 1$ and costs $c_{b,l,m} = \|d_{b,\cdot} - m \cdot l\|$ for an arbitrary norm $\|\cdot\|$, our heuristic SFA produces an optimal solution whenever all lot-types $l \in \mathcal{L}$ are checked.*

**Proof:** Since we loop over all $l \in \mathcal{L}$, the optimal lot-type is checked at some point. Thus we may assume that we are in the case where the optimal lot-type $l$ is chosen.

Without the total capacity constraint assigning to each branch $b$ its locally optimal multiplicity $m^\star(b)$ would be globally optimal. Now let us assume that the number of items $I$ in our assignment is larger than the upper bound $\overline{I}$ of the total capacity constraint. Let us have a look at an optimal assignment $\tilde{m}$ of the multipliers. If there exist two branches $b_1, b_2 \in \mathcal{B}$ such that $\tilde{m}(b_1) < m^\star(b_1)$ and $\tilde{m}(b_2) > m^\star(b_2)$ then increasing $\tilde{m}(b_1)$ and decreasing $\tilde{m}(b_2)$ by one yields another solution where the costs do not increase. Thus there exists an optimal solution where no multiplier of a branch is larger then the locally optimal multiplier. This optimal solution arises from our initial distribution plan (whenever it violates the total capacity constraint) by deleting exactly $\left\lceil \frac{I - \overline{I}}{\|l\|_1} \right\rceil$ lot packages. Due to the convexity of the norm function the greedy way of deleting lot packages of our heuristic ends up with this optimal solution. □

Since the adjustment step can be performed very fast, one might also take some kind of generalized swapping techniques into account. Because for these techniques there exists an overboarding amount of papers in the literature, we will not go into detail here, but we would like to remark that in those cases (see Subsection 4.5), where the optimality gap of our heuristic lies above 1 % swapping can improve the solutions of our heuristic by a large part.

## 4.5. *Experimental optimality gap of SFA*

In order to substantiate the usefullness of our heuristic, we have compared the quality of the solutions, given by this heuristic after one second of computation time (on a standard laptop: Intel® Core™ 2 CPU with 2 GHz and 1 GB RAM) with respect to the solution given by `CPLEX 11` (after solving to optimality).

Our business partner has provided us with historic sales information for nine different commodity groups, each ranging over a sales period of at least one-and-a-half years. From this we estimated mean demands via aggregating over products in a commodity group. By normalizing the lengths of the products' sales periods to the point in time when half of the product was sold out, we were able to mod out the effects of any product's individual success or failure. Prior to each test calculation, the resulting demands were scaled so that the total mean demand was in the center of the total capacity interval given by the management for a new order of a product in that commodity group.

For each commodity group we have performed a test calculation for $k \in \{2,3,4,5\}$ distributing some amount of items to almost all branches. The crucial parameters are given in Table 1, the results are presented in Table 2.

| Commodity group | $\lvert\mathcal{B}\rvert$ | $\lvert\mathcal{S}\rvert$ | $[\underline{I},\overline{I}]$ | $\lvert\mathcal{L}\rvert$ | $M$ |
|---|---|---|---|---|---|
| 1 | 1119 | 5 | [10 630, 11 749] | 243 | 10 |
| 2 | 1091 | 5 | [10 000, 12 000] | 243 | 10 |
| 3 | 1030 | 5 | [9 785, 10 815] | 243 | 10 |
| 4 | 1119 | 5 | [10 573, 11 686] | 243 | 9 |
| 5 | 1175 | 5 | [16 744, 18 506] | 243 | 15 |
| 6 | 1030 | 5 | [11 000, 13 000] | 243 | 9 |
| 7 | 1098 | 5 | [15 646, 17 293] | 243 | 9 |
| 8 | 989 | 5 | [11 274, 12 461] | 243 | 9 |
| 9 | 808 | 5 | [9 211, 10 181] | 243 | 10 |

Table 1. Parameters for the test calculations.

| Commodity group | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|
| 1 | 2.114 % | 1.226 % | 2.028 % | 1.983 % |
| 2 | 0.063 % | 0.052 % | 0.006 % | 0.741 % |
| 3 | 0.054 % | 0.094 % | 0.160 % | 0.170 % |
| 4 | 0.019 % | 0.007 % | 0.024 % | 0.038 % |
| 5 | 0.015 % | 0.017 % | 0.018 % | 0.019 % |
| 6 | 0.018 % | 0.022 % | 0.024 % | 0.022 % |
| 7 | 0.013 % | 0.013 % | 0.014 % | 0.014 % |
| 8 | 0.016 % | 0.017 % | 0.018 % | 0.019 % |
| 9 | 0.011 % | 0.939 % | 0.817 % | 0.955 % |

Table 2. Optimality gap in the $\|\cdot\|_1$-norm for our heuristic on nine commodity groups and different values for the maximum number $k$ of used lot-types.

We have identified two aspects preventing our heuristic to reach an optimal solution for larger $k$. If the number $k$ of applicatory lot-types increases, then obviously the number of $k$-subsets of $\mathcal{L}$ increases on a larger scale. In some cases the optimal $k$-subset hides from our heuristic and the time limit is reached before the optimal subset is checked. If the time limit is relaxed to 2 seconds then for commodity group 9 we obtain gaps of $0.011\,\%$ and of $0.012\,\%$ for $k=3$ and $k=4$, respectively. For $k=5$ the gap drops to $0.418\,\%$ after 2.4 seconds, to $0.250\,\%$ after 7.2 seconds, and to 0.114 % after 18.2 seconds. For commodity group 2 and $k = 5$ there remains a gap of 0.472 % after 9.2 seconds and a gap of 0.044 % after 41.4 seconds.

In case of commodity group 1 the large gaps are due to a heavy violation of the total capacity constraint for the best scored lot-type combinations. Here some kind of general swapping techniques significantly help to improve the solution below a gap of 1 % within the time limit.

Besides these nine test calculations we have done several calculations on our data sets with different parameters, we have, e.g., considered case with fewer sizes, fewer branches, smaller or larger intervals $[\underline{I}, \overline{I}]$, larger $k$, or larger sets $\mathcal{L}$ of applicatory lot-types. The results are from a qualitative point of view more or less the same as for the presented test calculations.

## 5. Conclusion and outlook

Starting from a real world optimization problem we have formalized a new general optimization problem, which we call the lot-type design problem.

In Subsection 3 we have given an integer linear programming formulation which has a very strong LP-relaxation on our instances. Although this approach is quite fast (computing times below half an hour), there was a practical need for fast heuristics to solve the problem. We have presented the heuristic SFA, which performs very well on real world data sets with respect to the optimality gap.

The question of a good approximation algorithm for the lot-type design problem is left as an open problem.

For the practical problem the uncertainties and difficulties concerning the demand estimation have to be faced. There are several ways to make solutions of optimization problems more robust. One possibility is to utilize robust optimization methods. Another possibility is to consider the branch- and size dependent demands as stochastic variables and to utilize integer linear stochastic programming techniques. See, e.g., [2] or more specifically [11]. These enhanced models, however, will challenge the solution methods a lot, since the resulting problems are of a much larger scale than the one presented in this paper. Nevertheless, this is exactly what we are looking at next.

As a vision, we would like to consider integrated models containing both the aspects of mark-down optimization and assortment planning and the lot-type design problem studied in this paper.

Finally, we report that thanks to very positive results in field tests conducted by our industry partner, a prototype implementation the SFA heuristic is currently in operation for generating the lot-type design of almost all current orders.

## Acknowledgements

This research was supported by a grant of the *Bayerische Forschungsstiftung, Az-727-06*. We would like to thank the anonymous referees for many helpful comments to improve the presentation of the paper.

## References

[1] Pasquale Avella, Antonio Sassano, and Igor Vasil'ev, *Computational study of large-scale p-median problems*, Mathematical Programming **109** (2007), no. 1, 89–114.

[2] J. R. Birge and Louveaux F., *Introduction to stochastic programming*, Springer Series in Operations Research and Financial Engineering, Springer, 1997.

[3] Maurizio Boccia, Antonio Sforza, Claudio Sterle, and Igor Vasilyev, *A cut and branch approach for the capacitated p-median problem based on fenchel cutting planes*, Journal of Mathematical Modelling and Algorithms **7** (2007), no. 1, 43–58.

[4] Alberto Ceselli and Giovanni Righini, *A branch-and-price algorithm for the capacitated p-median problem*, Networks **45** (2005), no. 3, 125–142.

[5] M. Charikar, S. Guha, E. Tardos, and D. Shmoys, *A constant-factor approximation algorithm for the k-median problem*, 31st Annual ACM Symposium on the Theory of Computing, 1999, pp. 1–10.

[6] Wedad Elmaghraby and Pinar Keskinocak, *Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions*, Management Science **49** (2003), 1287–1309.

[7] A. Gürhan Kök and Mashall L. Fisher, *Demand estimation and assortment optimization under substitution: Methodology and application*, Operations Research **55** (2007), no. 6, 1001–1021.

[8] Nenad Mladenović, Jack Brimberg, Pierre Hansen, and José A Moreno-Pérez, *The p-median problem: a survey of metaheuristic approaches*, Eur. J. Oper. Res. **179** (2007), no. 3, 927–939.

[9] Yves Pochet and Laurence A. Wolsey, *Production planning using mixed integer programming*, Springer, 2006, ISBN 978-0-387-29959-4, 477 pages.

[10] Mauricio G.C. Resende and Renato F. Werneck, *A hybrid heuristic for the p-median problem*, J. Heuristics **10** (2004), no. 1, 59–88.

[11] C. Swamy, *Approximation algorithms for clustering problems*, Ph.D. thesis, Cornell University, 2004.

[12] Laurence A. Wolsey, *Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation*, Management Science **48** (2002), no. 12, 1587–1602.