

The combinatorics of S, M, L, XL: The best fitting delivery of T-shirts

Constantin Gaul¹, Sascha Kurz¹, and Jörg Rambau¹

University of Bayreuth, Dep. of Mathematics, Physics, and Computer Science
95440 Bayreuth, Germany
`firstname.lastname@uni-bayreuth.de`

Abstract. 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

1 Introduction

The problem studied in this note is motivated by a special feature of the ordering process of a fashion discounter with many branches: 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 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. The number of different lot-types is bounded by the supplier.

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)*. A detailed version of this work appeared in [1], where also references to related work can be found.

* This research was supported by a grant of the *Bayerische Forschungsstiftung*, Az-727-06.

2 The lot-type design problem

Formally, the problem can be stated as follows: 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{L}, \bar{L} 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.

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{L}, \bar{L} 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 $[\underline{L}, \bar{L}]$
(the *total capacity constraint*)
- the distance of the order from the demand measured by $\|\cdot\|$ is minimal

The LDP can be formulated as an Integer Linear Program if we restrict ourselves to the L^1 -norm for measuring the distance between supply and demand. This norm is quite robust against outliers in the demand estimation.

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.

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

$$s.t. \quad \sum_{l \in \mathcal{L}} \sum_{m=1}^M x_{b,l,m} = 1 \quad \forall b \in \mathcal{B} \quad (2)$$

$$\sum_{l \in \mathcal{L}} y_l \leq k \quad (3)$$

$$\sum_{m=1}^M x_{b,l,m} \leq y_l \quad \forall b \in \mathcal{B}, \forall l \in \mathcal{L} \quad (4)$$

$$\underline{I} \leq \sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{L}} \sum_{m=1}^M \sum_{s \in \mathcal{S}} m \cdot l_s \cdot x_{b,l,m} \leq \bar{I} \quad (5)$$

$$x_{b,l,m} \in \{0, 1\} \quad \forall b \in \mathcal{B}, \forall l \in \mathcal{L}, \forall m = 1, \dots, M \quad (6)$$

$$y_l \in \{0, 1\} \quad \forall l \in \mathcal{L} \quad (7)$$

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

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.

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 briefly sketch the idea of the heuristic *Score-Fix-Adjust (SFA)*: It

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).

Details can be found in [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 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.*

In order to substantiate the usefulness 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	$ \mathcal{B} $	$ \mathcal{S} $	$[\underline{L}, \bar{L}]$	$ \mathcal{L} $	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.

We can see that – given the uncertainty in the data – the performance of SFA is more than satisfactory.

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.

3 Conclusions

We identified the lot-type design problem in the supply chain management of a fashion discounter. It can be modeled as an ILP, and real-world instances can be solved by commercial-of-the-shelf software like CPLEX in half an hour whenever the number of lot-types is not too large.

Our SFA-heuristics finds solutions with a gap of mostly under 1 % in a second, also for instances with a large number of lot-types. Given the volatility of the demand estimation, these gaps are certainly tolerable.

Meanwhile, the model and SFA have been put to operation by our business partner with significant positive monetary impact.

References

1. Gaul, C., Kurz, S., Rambau, J.: On the lot-type design problem. Optimization Methods and Software (2009) <http://www.informaworld.com/10.1080/10556780902965163>.