Space optimization in retail using a Multiple Choice Multiple Knapsack Problem solver

David Pisinger

August 15, 2014

Abstract

The Space optimization problem in retail is defined as follows: Given a set of departments and a set of fixtures. Each department can be realized in a number of sizes (widths) each size having a different profitability. The profitability can be measured in sales, turnaround, or net profit. The space optimization problem now has to assign departments to fixtures and choose a size for each department, so that the overall length of each fixture is respected, and so that the overall profitability is maximized. A valid solution, furthermore, has to satisfy a number of business rules: A department may have a mandatory position, or a department may be optional (i.e. it can have size zero), a department may have forbidden positions. Moreover, some departments may need to be adjacent in order to ensure shoppability.

We show how the space optimization problem may be defined mathematically as a multiple-choice multiple knapsack problem with additional constraints. The additional constraints describe the extra business rules, i.e. mandatory/forbidden positions, adjacency etc.

An algorithm is presented for solving the multiple-choice multiple knapsack problem with additional constraints. The algorithm is based on a branch-and-bound approach, assigning departments to fixtures. Once all departments have been assigned to a fixture, the size of each department can be found by solving a multiple-choice knapsack problem for each fixture. Upper bounds are found by Lagrangian relaxing all capacity constraints. Symmetries in the solution space are broken by use of lexicographical orderings. Tightening of the bounds is performed by enumerating all size combinations of departments, for departments currenly assigned to a given fixture. This can be done quite efficiently by use of dynamic programming. Combinations of department sizes that exceed the capacity of a fixture are deleted leading to a tightened upper bound.

Computational results are reported for real-life space optimization problems. Problems with up to 25 departments and 10 fixtures, having a large number of business rules, are solved in a few seconds.

1 Introduction

Benjamin Franklin introduced the phrase "time is money". In retail one should perhaps rather use the phrase "space is money". Having more space in a department makes it possible to offer more brands, and some brands may even have more facings, making the brand more visible. However, space is expensive, and most stores have a fixed amount of space available that cannot be extended. This makes it attractive to optimize the space utilization, so that every inch of space is used in the best way possible.

The Space optimization problem is defined as follows: Given a set of departments and a set of fixtures. Each department can be realized in a number of sizes (widths) each size having a different profitability. The profitability can be measured in sales, turnaround, or net profit. The space optimization problem now has to assign departments to fixtures and choose a size for each department, so that the overall length of each fixture is respected, and so that the overall profitability is maximized. A valid solution, furthermore, has to satisfy a number of business rules: A department may have a mandatory position, a department may be optional (i.e. it can have size zero), a department may have forbidden positions. Moreover, some departments may need to be adjacent in order to ensure shoppability.

1.1 Literature

The space optimization problem in retail is closely related to the multiple-choice multiple knapsack problems (MCMKP). In the MCMKP one has several disjoined classes of items, and a number of knapsacks. Each item has a profit and weight. The task is to choose exactly one item from each class and assign it to a knapsack so that the overall profit is maximized while respecting the capacity on the weight for each knapsack.

In the literature MCMKP is used to denote a number of different problems. The most general model of the multiple-choice multiple knapsack problem is

$$\text{maximize} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k \in N_j} p_{ijk} x_{ijk} \tag{1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} \sum_{k \in N_j} w_{ijk} x_{ijk} \leq c_i, \quad i = 1, \ldots, m, \tag{2}$$

$$\sum_{i=1}^{m} \sum_{k \in N_j} x_{ijk} = 1 \qquad j = 1, \ldots, n, \tag{3}$$

$$x_{ijk} \in \{0, 1\}, \qquad i = 1, \ldots, m, \; j = 1, \ldots, n, \; k \in N_j \tag{4}$$

Notice that the profit and weight of an item depends on the knapsack it is assigned to.

In the *flat profit* multiple-choice multiple knapsack problem, the profit of an item is the same whichever knapsack the item is assigned to. This means that the objective function in the above problem becomes

$$\text{maximize} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k \in N_j} p_{jk} x_{ijk} \tag{5}$$

In the *flat weight* multiple-choice multiple knapsack problem, the weight of an item is the same whichever knapsack the item is assigned to. This means that the capacity constraint (2) becomes

$$\sum_{j=1}^{n} \sum_{k \in N_j} w_{jk} x_{ijk} \le c_i, \quad i = 1, \ldots, m, \tag{6}$$

Not much work has been done on multiple-choice multiple knapsack problems, and the problem is not covered in the book on knapsack problems [7].

Various heuristics for the flat profit (but variable weight) problem have been proposed by [1], [3], [2], [5], [6] and [9]. Han et al. [4] also consider the flat profit problem, comparing various algorithms in systematically generated benchmark instances. They conclude that instances are hard to solve when profits are correlated with weights, and certain categories of instances are very hard for all considered algorithms. The authors expect symmetry to be a major challenge in solving MCMKP.

To the best of our knowledge, none of the mentioned papers consider the flat weight, variable profit case, and also they cannot handle the extra business constraints: mandatory fixture, forbidden fixture, adjacent departments. The latter constraints are, however, quite easy to implement in a branch-and-bound algorithm as will be shown.

1.2 Overview

In Section 2 we present a simple example from Walgreens Company to illustrate the problem of space optimization in retail. Section 3 describes the problem in details using a terminology from retail, while the following Section 4 introduces the knapsack teminology used in the sequel of the paper. Section 5 introduces the overall branch-and-bound algorithm, discussing branching strategies, branching order, and various techniques for breaking symmetries in the solution space. Section 6 shows how upper bounds of high quality can be found in linear time by Lagrangian relaxing the capacity constraints. Section 7 introduces the (single knapsack) multiple-choice knapsack problem that is used in the butoom of the branch-and-bound algorithm to find the size of each department, and Section 8 discuss how the same problem can be used to tighten upper bounds through partial enumeration. The paper is concluded by some preliminary results in Section 9.

2 Case Study

In order to illustrate the problem, we consider the following problem from Walgreens Co. A pharmacy corner has 10 fixtures as shown on Figure 1. The length and height of the fixtures is described in Table 2.

We have 17 departments as described in Table 3. Each department has to be assigned to a fixture, and it should be decided how much space the department should have. Obviously, assigning more space to a department resluts in higher profit (measured in various ways). Estimates for the

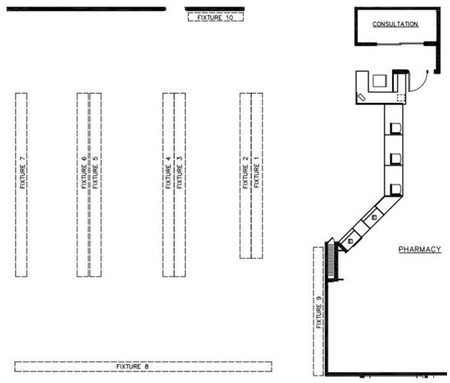

Figure 1: Pharmarcy corner with 10 fixtures. Fixtures 1 to 8 are counter fixtures with height 66 in., while fixtures 9 and 10 are standard wall fixtures with height 72 in.

expected profit are given in Table 1. Each department can be realized in steps of 3 inches from a lower limit up to an upper limit. Depending on the space assigned to a department, different profits can be expected.

Not all departments can be assigned to each fixture. Weight constraints or visual appearance can restrict which departments can be assigned to a given fixture. This is illustrated in Table 4, where a department can have a mandatory fixture, or a department can have some fobidden fixtures.

It is important that the assignment of departments to fixtures is logical, so that customers easily can find the relevant products. The shoppability constraints are defined in Table 3. The constraints can state that two departments should be placed next to each other, since customers typically will buy these products together.

3 Problem description

In order to describe the problem more formally we introduce the following notation:

- Let m be the number of fixtures, and c_i be the size of fixture $i = 1, \ldots, m$.
- Let n be the number of departments, each department $j = 1, \ldots, n$ having various choices for realiztion N_j. I.e. every choice $k \in N_j$ represents a possible space allocation of department j.
- Let p_{jk} be the profit of department j having choice $k \in N_j$
- Let w_{jk} be the size of department j having choice $k \in N_j$
- M_j be the set of mandatory fixtures for department j (obviously $|M_j| \leq 1$)
- F_j be the set of forbidden fixtures for department j

dept ID	length	profit
3301	3	768
3301	6	997
3301	9	1017
3301	12	1017
3301	15	1017
3303	3	275
3303	3	472
3303	3	572
3303	12	608
3303	15	618
3303	18	619
3303	21	620
3315	3	425
3315	6	651
3315	9	741
⋮	⋮	⋮

Table 1: Profit table for fixtures having height 66 in. For instance department 3303 can be between 3 inches and 21 inches long. If 12 inches are assigned to the department then a profit of 608 is expected.

fixture ID	length (in)	type	height (in)
1	27	counter	66
2	27	counter	66
3	30	counter	66
4	30	counter	66
5	30	counter	66
6	30	counter	66
7	30	counter	66
8	39	counter	66
9	21	standard wall	72
10	9	standard wall	72

Table 2: Overview of available fixtures. For instance fixure 8 is a counter fixture having length 39 inches, and height 66 inches.

dept ID	description	adjacency dept ID
3301	smoke buster	
3303	foot needs	
3315	elastics	3369 or 3365
3325	incontinence	
3330	eye ear	
3351	Rx diagnostics	3301
3352	pill minder	
3353	vitamins	
3364	pediatrics	3367 or 3370
3365	convalescence	
3366	antacid laxative	
3367	cough cold	
3368	diet sports nutrition	
3369	first aid	
3370	pain sleep	
3381	medical nutrition	
6399	wellness vitamins	

Table 3: Department adjacency. For instance dept 3351 needs to be adjacent to dept 3301, and dept 3315 needs to be adjacent to either dept 3369 or dept 3365

dept ID	description	mandatory	forbidden location(s)
3301	smoke buster		
3303	foot needs		1,2,4,5,6,7,8
3315	elastics		
3325	incontinence		9,10
3330	eye ear		
3351	Rx diagnostics		
3352	pill minder		
3353	vitamins		
3364	pediatrics		
3365	convalescence		9,10
3366	antacid laxative		
3367	cough cold	8	
3368	diet sports nutrition		
3369	first aid		
3370	pain sleep		
3381	medical nutrition		
6399	wellness vitamins		

Table 4: Department/fixture compatibility. For instance dept 3367 must be assigned to fixture 8, while dept 3325 can be assigned to all fixtures except fixture 9 or 10

- A_j be the set of departments that must be adjacent to department j (at least one of the departments in A_j should be adjacent to j)

Notice that if a department j is *optional* it can be handled by extending the set N_j of possible realizations with the choice $(0,0)$. This will make it possible to choose an empty assignment, and hence in practice omitting the department.

In order to formulate the problem as an IP model we introduce the binary variables x_{ijk} where $x_{ijk} = 1$ iff choice k of department j is assigned to fixture i. This leads to the model: