A heuristic algorithm for space allocation in a pallet storage warehouse

Patrik Norén & Johan Eriksson

The model described in this article is a proposed solution to a problem that came up in the authors' master thesis; Inventory control and warehouse management at Kraft Foods production unit for chocolate, written at Lund Institute of Technology, Faculty of Engineering Logistics, Department of Industrial Management and Engineering. In this article, the problem as well as the solution method is generalized as much as possible in order to simplify for readers who want to apply the model to the problems they are facing.

Introduction

Two ways of keeping track of where to find a specific good (material) in a pallet storage warehouse are recognized; working with fixed storage spaces or having a computer based system that is constantly updated with the content at each storage space. Anyhow, one would like to have the cost for material handling as low as possible. Most often the cost is driven by labor, and in this case by time spent by fork lift drivers on material handling. It is hence natural to try to minimize handling time. When working with *the method of fixed storage locations* for each material, it is possible to minimize the handling time by dedicating the storage locations to the different materials in the most proper way.

The heuristic model presented in this paper seeks to, if not minimize, at least to decrease the time of material handling in warehouses. The benefits with the model are that it is easy to use in practice and that the data used in the model often is found easily in a company's ERP-system.

The Environment

In certain production environments pallet goods is often stored in *pallet racks* in the same building as, and fairly close to, the production lines. In such production environments different goods have a slightly different flow, meaning that different materials may be used at different locations along the production lines. The different flows can yet be said to be made up by four standard handling operations: to bring a pallet out and down from its storage location in a pallet rack, to bring a pallet up and into its storage location in a pallet rack, to transport a pallet of material from its storage location to where it is used in the production, to transport a pallet from the goods reception to a storage location.

Optimal formulation

Minimizing the total handling time is by definition the same as minimizing the combined time of all operations that makes up the handling time:

$$\min \textit{handling time} = \min(\textit{Operation time } 1 + \textit{Operation time } 2 + \cdots + \textit{Operation time } n)$$

Let us introduce the following notation:

M	set of all materials m
L	set of all storage spaces l
t_l^d	time to bring a pallet out and down from storage location l in a pallet rack
t_l^u	time to bring a pallet up and into storage location l in a pallet rack
t_{lm}	time to transport a pallet of material m from storage location l to where it is used in the production, not including out and down/up and in
t_l^g	time to transport a pallet from the goods reception to storage space l
d_i	number of times per year that operation i is executed for material m
s_m	number of storage spaces dedicated to material m

Let X_{lm} be a binary variable according to:

$X_{lm} = 1$	storage space l belongs to material m
$X_{lm} = 0$	storage space l does not belong to material m

Before the problem can be set up, some basic assumptions have to be made**:**

- The number of storage spaces that a material should have is decided at an earlier stage, s_m.
- The fact that the number of pallets in stock for a material can be larger than s_m is disregarded. If this is not negligible, the number of dedicated storage spaces should be reconsidered.
- That all storage locations dedicated to one material are used equally on average.
- The time spent on the basic handling operations t_l^d, t_l^u and t_{lm} is equal regardless of which material that is handled.

Also incorporate the notation $t_i^{lm} = (a_i * t_l^h + b_i * t_l^n + c_i * t_l^u + d_i * t_{lm})$ where a_i to d_i simply denotes how many of each basic handling operation that is required for each handling operation o_i.

The minimization problem can then be constructed as a BIP-problem (Binary Integer Programing):

$$\min \sum_l \sum_m \frac{X_{lm} * o_1 * t_1^{lm}}{s_m} + \frac{X_{lm} * o_2 * t_2^{lm}}{s_m} + \cdots + \frac{X_{lm} * o_n * t_n^{lm}}{s_m} \qquad (1)$$

under the constraints:

$$\sum_m X_{lm} \leq 1 \qquad \forall\, l \in L \qquad (2)$$

$$\sum_l X_{lm} = s_m \qquad \forall\, m \in M \qquad (3)$$

The first constraint, equation (2), implies that each storage location can be dedicated to no more than one material. The second constraint, equation (3), makes sure that the correct number of storage locations is given to each material.

A pure mathematical optimization of this problem is very complex and rapidly growing in size. For a warehouse with 60 different materials and 200 storage locations the problem will have 12 000 binary variables. The number of possible combinations is then:

$$2^{12000} \approx 2{,}3 * 10^{3612}$$

This is too many for a computer to evaluate all within a reasonable time and therefore a heuristic optimization procedure is described, which is intuitive and easy to follow, and thus can be applied by a wider group of users. By slightly rewriting equation (1) and removing the binary variables (and hence also the summation notations) the following expression is obtained:

$$\frac{o_1 * t_1^{lm}}{s_m} + \frac{o_2 * t_2^{lm}}{s_m} + \cdots + \frac{o_n * t_n^{lm}}{s_m} =$$

$$\frac{o_1 * \overline{t_1}}{s_m} + \frac{o_2 * \overline{t_2}}{s_m} + \cdots + \frac{o_n * \overline{t_n}}{s_m} =$$

$$\frac{\boldsymbol{o_1 * w_1 + o_2 * w_2 + \cdots + o_n * w_n}}{\boldsymbol{s_m}} \qquad (4)$$

For each material, this gives a measure of how often its storage location on average is involved in a handling operation, weighted by the average time it takes to perform each of the handling operations, o_i. This is then also a measure of how much a material drives cost in terms of the time it is handled. The measure is therefore considered reasonable to rank the materials after. By using the thought of minimizing the handling time for the largest cost drivers is most important, the following algorithm was developed:

1. Calculate the number of different handling operations for each material and estimate the time it takes to perform them each.
2. Determine the number of seats to be allocated to a particular material.
3. Rank the materials from a list of weighted handling times per allocation of storage locations, according to equation (4).
4. Assign the most “attractive” locations to the material at the top of the ranking list. (Attractive meaning the ones that intuitively would imply the least handling time for the specific material)
5. Remove the material at the top of the ranking list and mark its storage locations as occupied.
6. If the list is not empty, go back to Step 4.
7. Make the vacant locations that are left over so-called shared storage locations.

The algorithm is very similar to the "greedy" algorithm that Danzig (1957) uses to give an approximate solution to the well-known Knapsack problem[1]. The algorithm is also similar to the so-called first-fit algorithm[2], a well-known heuristic for so-called bin-packing problems, which, inter alia, is described by Vazirani (2003). Both of these heuristics has been shown to have a worst performance that is *M/2* where *M* is the best solution. In this case the meaning is that the worst solution the algorithm can give is that the time spent on material handling is twice as high as optimal. However, it is very unlikely that the algorithm gives to such poor results. Danzig also suggests that the human brain seems to have an incredible capacity to scan many

[1] Danzig, G. (1957), *Discrete-Variable Extremum Problems*.
[2] Vazirani, V (2003), *Approximation Algorithms.*

different combinatorial solutions and quickly come up with a very good one, although the number of possible combinations is very large. It is therefore recommended to always scan the solution provided by the algorithm manually, in order to see if switching storage locations for a few materials can save handling time. The scan can also be done through so called 2-opt[3], a local search optimization algorithm first proposed by Croes in 1958.

A practical advantage of the method described is that the different handling operations, o_i, mostly often can be obtained from a company's ERP-system. A disadvantage that the method has is its sensitivity to choice of "average times", meaning that going from t_i^{lm} to $\overline{t_\iota}$ when reaching equation (4) implies assuming an average time it takes to perform each operation regardless of which material and storage locations are used. This can be very hard, but in practice it is inly important that the estimates are correct relative to time for other operations, as the weights calculated can all be rescaled without changing the final result.

Conclusions

The model in this paper is a practical and easy model that can help companies to improve their warehouse management by placing material in a way that decreases the time consumed to handle pallets. It has some limitations, e.g. that it is assumed that the handling of all pallets take the same amount of time and the model does not necessary place every pallet in their optimal place but as shown above it is often practically impossible. Considering the little amount of time that needs to be spent in order to collect the necessary data and the relatively big difference an implementation of this model could mean for a company, we believe that it could ease the everyday life at a lot of companies.

References

Vazirani, V. (2003) *Approximation Algorithms.* Berlin: Springer.

Danzig, G.B. (1957), *Discrete-Variable Extremum Problems*. Operations Research, Vol5, No2

Croes, G.A. (1958) *A method for solving traveling-salesman problems.* Operations research Nov-Dec 1958.

[3] Croes, G.A. (1958) *A method for solving traveling-salesman problems.*