

Portuguese Study Groups' Reports

Report on
Warehouse storing and collecting of parts

Problem presented by GROHE at the
65th European Study Group with Industry
21st–24th April 2008
Centro de Matemática da Universidade do Porto
Portugal

October 13, 2008

Problem presented by: Jorge Novo (GROHE)

Study group contributors: Jorge Almeida, J. Orestes Cerdeira,
Manuel Cruz and Paulo B. Vasconcelos

Report prepared by: Jorge Almeida¹, J. Orestes Cerdeira²,
Manuel Cruz³ and Paulo B. Vasconcelos⁴

¹Centro de Matemática da Universidade do Porto and Dep. Matemática Pura, Faculdade de Ciências da Universidade do Porto, e-mail: jalmeida@fc.up.pt

²Centro de Estudos Florestais and Dep. Matemática, Instituto Superior de Agronomia, Technical University of Lisbon (TULisbon), e-mail: orestes@isa.utl.pt

³Instituto Superior de Engenharia, Instituto Politécnico do Porto, e-mail: mbc@isep.ipp.pt

⁴Centro de Matemática da Universidade do Porto and Faculdade de Economia da Universidade do Porto, e-mail: pjv@fep.up.pt

Abstract

This report deals with reducing the high costs resulting from the wear and tear of the fork-lifts used to store or collect items in a warehouse. Two problems were identified and addressed separately. One concerns the way items should be stored or collected at storage locations on the shelves of one corridor. The other problem seeks for an efficient way to define which fork-lift should operate on each corridor, and the order by which the fork-lifts should visit the corridors. We give to both problems formulations that fit in the framework of combinatorial optimization.

1 Introduction

GROHE is a manufacturer and supplier of sanitary fittings. GROHE presented a specific operating planning problem on one of the company's warehouses in Portugal.

Warehouse operating planning is a sensible matter given the large contribution of this sector to the total costs of many companies. This justifies the intense work directed to this subject. We refer to [4] for a recent extensive review of the topic, linking views from academic researchers and warehouse practitioners.

GROHE is interested in reducing the costs resulting from storing and collecting items, such as traps, waste traps, flow straighteners, . . . , in the warehouse.

The warehouse consists of several corridors of shelves crossed by a perpendicular passageway as depicted in Figure 1.1.

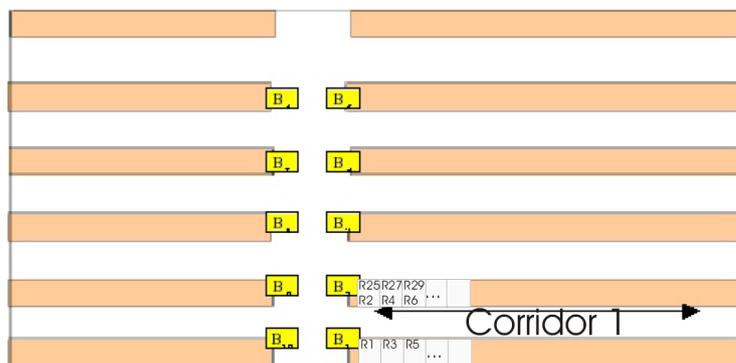


Figure 1.1: Warehouse template.

The reports with the orders for storing or collecting items, together with the location of each item in the warehouse, are given in the format (spreadsheet style) shown in Figure 1.2, specifying the *flow* (in - store or out - collect), and the storage locations where the items should be placed or collected. Each storage location is identified by the corridor number, the shelf's level, the rack number, and the bin number.

The items that shall be stored or collected at storage locations of the same corridor will be referred as *corridor's items*.

Storage and collecting operations are accomplished by three fork-lifts that operate inside the corridors. Each fork-lift cannot carry more than one item in each career. Hence, a career consists of:

- a) collecting an item from a depot located near the corridor, carrying it to a store position i , move to some store position j to pick another item, and take it to the depot; or

flow	corridor	rack	floor	bin	bin nr.
in	02	18	07	2	02.18.07.2
in	02	01	01	2	02.01.01.2
in	02	04	08	3	02.04.08.3
in	02	21	05	3	02.21.05.3
in	02	20	02	3	02.20.02.3
in	02	11	02	1	02.11.02.1
in	02	02	01	2	02.02.01.2
in	02	10	02	1	02.10.02.1
in	02	23	07	2	02.23.07.2
in	02	14	05	1	02.14.05.1
in	02	05	02	3	02.05.02.3
out	02	8	08	3	02.08.08.3
out	02	01	05	3	02.01.05.3
out	02	10	01	2	02.10.01.2
out	02	03	03	2	02.03.03.2
out	02	14	07	3	02.14.07.3
out	02	16	03	1	02.16.03.1
out	02	08	01	3	02.08.01.3
out	02	11	05	1	02.11.05.1
out	02	21	07	3	02.21.07.3
out	02	16	05	2	02.16.05.2
out	02	16	03	2	02.16.03.2
out	02	03	02	3	02.03.02.3
out	02	07	04	2	02.07.04.2

Figure 1.2: A typical working order from the company.

- b) carry an item from the depot to a store position i , and come back to the depot; or
- c) leaving the depot unloaded to collect an item at storage location j and return to the depot.

It should be noted that, if the storage locations i and j are not on the same side of the corridor, the careers of type a) require some extra work. Indeed, once inside the corridor, the fork-lift can only work on one side. To operate on the opposite side requires that the orientation of the forks has been previously reversed, which has to be done outside the corridor.

Let us call *corridor task* a feasible assignment of the corridor's items to careers a), b) or c).

The greatest concern of the company is to reduce the high costs due to the wear and tear of the fork-lifts. This is directly proportional to their

usage and is mostly determined by the distances covered on the store/collect operations. The company believes that costs could be significantly reduced if the corridor tasks were carefully settled. This was by far considered the most important question of the store/collect planning.

Additionally, the company would like to find an efficient way to define which fork-lift should operate on each corridor, and the order by which the fork-lifts should visit the corridors.

We address these issues in Sections 2 and 3, and finish this report with some comments about the layout optimization in Section 4.

2 Optimization within one corridor

Here we show how to determine optimal corridor tasks.

Let S and C denote the sets of the storage locations of the corridor's items that are going to be stored and to be collected, respectively.

If an artificial store position, say 0, is used to refer the depot located near the entrance of the corridor, every career can be identified with an ordered pair of "store positions" (i, j) , that will be

- from type a) if $i \in S$ and $j \in C$,
- from type b) if $i \in S$ and $j = 0$,
- from type c) if $i = 0$ and $j \in C$, or
- an *empty career* if $i = j = 0$.

As previously referred the main concern is the lifetime of the fork-lifts. To estimate the wear and tear resulting from the store/collect operations, a number of features should be taken into consideration, including distances and differences between the shelf levels of pairs of storage locations, and the extra amount of work required for pairs of locations on opposite sides of the corridor.

Let c_{ij} be the *cost* estimated for the pair of positions (i, j) , and define $c_{00} = 0$ to be the cost of the empty career.

We now extend sets S and C with as many copies of the "store position" 0, so that the resulting collections, denoted by \bar{S} and \bar{C} , respectively, have both size $|S| + |C|$.

Note that we can identify the feasible corridor tasks with the bijections $\pi : \bar{S} \rightarrow \bar{C}$. The corridor task corresponding to π is the set of careers $T_\pi = \{(i, \pi(i)), \text{ with } i \in \bar{S}\}$. Moreover, the cost of the corridor task T_π is $c(\pi) = \sum_{i \in \bar{S}} c_{i\pi(i)}$.

To illustrate this construction, consider the fictitious locations sets $S = \{a, b\}$ and $C = \{c, d, e\}$. The resulting collections are $\bar{S} = \{a, b, 0, 0, 0\}$ and $\bar{C} = \{c, d, e, 0, 0\}$, and Table 1.1 shows how the costs extend to \bar{S} and \bar{C} .

	c	d	e	0	0
a	c_{ac}	c_{ad}	c_{ae}	c_{a0}	c_{a0}
b	c_{bc}	c_{bd}	c_{be}	c_{b0}	c_{b0}
0	c_{0c}	c_{0d}	c_{0e}	0	0
0	c_{0c}	c_{0d}	c_{0e}	0	0
0	c_{0c}	c_{0d}	c_{0e}	0	0

Table 1.1: Costs of careers (i, j) , for $i \in \bar{S} = \{a, b, 0, 0, 0\}$ and $j \in \bar{C} = \{c, d, e, 0, 0\}$.

Hence, to search for an optimal corridor task reduces to finding π that minimizes $c(\pi)$. This is a well-known combinatorial optimization problem, the *assignment problem*, which can be polynomially solved, for instance, by the ‘‘Hungarian algorithm’’ [6, 8].

A MATLAB⁵ implementation is available at the end of this report. In order to be able to simulate an execution of the code a fictitious cost matrix was created, based on the Euclidean distance of the storage locations. Figure 1.3 shows the result for the provided working order (Figure 1.2), where the unpaired items appear at the end along with the ‘‘N/A’’ (Not-Available) mention.

odd rack		corridor 2	even rack	
02.01.01.2	02.03.02.3		02.18.07.2	02.16.05.2
02.21.05.3	02.11.05.1		02.04.08.3	02.08.08.3
02.11.02.1	02.07.04.2		02.20.02.3	02.16.03.1
02.23.07.2	02.21.07.3		02.02.01.2	02.08.01.3
02.05.02.3	02.03.03.2		02.10.02.1	02.10.01.2
N.A.	02.08.08.3		02.14.05.1	02.14.07.3
			N.A.	02.08.01.3

Figure 1.3: The output from an order storing/collecting.

Before tackling the routing problem for the warehouse, we may suggest to consider settling a 2-level buffer shell at each depot. This would facilitate the interface between fork-lifts and the vehicles bringing and collecting items from the depot. However, an overlapping between the two types of vehicles may have to be addressed.

⁵MATLAB is a trademark and product of The MathWorks, Inc.

3 Optimization in the warehouse

In this section we address the problem of assigning fork-lifts to corridors and settling how each fork-lift should visit the corridors assigned to it.

Let D denote the place in the warehouse where the fork-lifts stand when they are not operating. We will refer to D as the *depot*. Let N be the set that includes D and every corridor where a storage or collecting operation will take place. For $i, j \in N$, let d_{ij} be the *distance* estimated in terms of the wear and tear experienced by the fork-lifts when they move directly from i to j .

If only a fork-lift existed, the problem would be the *traveling salesman problem* [7, 1], which asks for a route of minimum total distance that starts in D , visits every other point of N exactly once, and then returns to D . The traveling salesman problem (TSP) is NP-hard [3] and, perhaps, it is the most studied problem in combinatorial optimization.

When more than one vehicle is available at the depot and no constraint exists on the size of the route of each vehicle, the problem is the multi-traveling salesman problem, which can be easily reduced to the (single) TSP.

If we modeled the problem as a multi (3 fork-lifts) TSP, optimal solutions would probably include very unbalanced routes. Indeed, since the times spent working inside the corridors are not considered, solutions of small total distances would most certainly assign long working periods to one (or two) fork-lift while the other(s) operates only a short (or even no) period (at all). Solutions would therefore tend to require excessively long periods of time to complete all the store/collect operations.

We propose a way to incorporate, within the vehicle routing framework, the working times spent inside the corridors, and which allows to balance the routes in order to reduce the total time.

The *capacitated vehicle routing problem* [9] is a generalization of the multi-TSP where the vehicles have limited carrying capacities to supply customers with known quantities of certain goods.

Our approach works as follows. Let us establish for each corridor (customer) in N a demand equal to the time estimated to execute the corridor task defined by the procedure of Section 2. Let T be a reasonable guess for the completion time of all the corridors tasks. Define the capacity of each of the three fork-lifts to be equal to $\frac{T}{3}$, and use one of the numerous algorithms for the capacitated vehicle routing problem [9] to run on this input.

If no feasible solution exists, the value of T should be increased. Otherwise, it may be considered the reduction of the value of T . In both cases run the algorithm again to check whether a feasible solution exists. This procedure can be repeated, using for instance the bisection method on T , until some satisfying solution is reached. Solutions obtained this way can combine small distances with limited total operation times, equitably divided by the

three fork-lifts.

4 Conclusions and recommendations

Layout design concerns the layout of the facility and the aisle configuration, factors that can significantly contribute to reduce costs. We strongly recommend a visit to the Erasmus-Logistica Warehouse website [10] and to the Interactive Warehouse website [11] where, along with an immense roll of bibliography on the subject, several on-line procedures can be interactively used to experiment and compare different layouts for several storage and routing strategies. We believe that a number of layout design options may be considered which may further help saving the wear and tear of the fork-lifts.

The company may also want to consider alternative strategies for the storage of items. There are several possibilities which may be seen in [5], for instance. Here we enumerate a few:

1. Random strategy: Items are allocated randomly. This is a strategy well suited for a computer-controlled warehouse. This policy may not be the best for the company since it allows a high space utilization but at the cost of increased travel distance.
2. Closest open location: Items are allocated at the first empty location resulting in a warehouse where racks are full around the depots.
3. Full-turnover scheme: Items are distributed as a function of their turnover. This is a good choice if items do not change frequently. Apparently the company can benefit from a combination of this scheme together with a dedicated strategy, where each item has a fixed location. The drawback is the reserved empty space for items that are out of stock and the advantages are the familiarity of order pickers with the locations and a suitable storage of items with different weights and sizes.
4. Family-grouping: Similar items or items which tend to be collected or stored together are allocated close together, preferably in the same corridor. This can greatly reduce fork-lifts usage.

Bibliography

- [1] D.L. Applegate, R.M. Bixby, V. Chvatal, W.J. Cook, *The Traveling Salesman Problem: a Computational Study*, Princeton University Press, Princeton, 2006.
- [2] R. Dekker, R. de Koster, K. J. Roodbergen, H. Van Kalleveen, Improving order-picking response time: time at Ankor's warehouse. *Interfaces* 34:4 (2004), 303-313.
- [3] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, W.H. Freeman & Company, San Franscisco, 1979.
- [4] J. Gu, M. Goetschalckx, L. F. McGinnis, Research on warehouse operation: a comprehensive review, *European Journal of Operational Research* 177 (2007) 1-21.
- [5] R. Koster, T. Le-Duc, Kees Jan Roodbergen, Design and control of warehouse order picking: A literature review, *European Journal of Operational Research* 182 (2007) 481-501.
- [6] Harold W. Kuhn, The Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, 2 (1955) 83-97.
- [7] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, *The Traveling Salesman Problem*, Ed. Chichester, John Wiley & Sons, 1985.
- [8] J. Munkres, Algorithms for the Assignment and Transportation Problems, *Journal of the Society of Industrial and Applied Mathematics*, 5 (1957) 32-38.
- [9] P. Toth, D. Vigo, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.
- [10] <http://www.fbk.eur.nl/OZ/LOGISTICA>.
- [11] <http://www.roodbergen.com/warehouse/>.
- [12] <http://www.mathworks.com/matlabcentral/>.

Appendix

```
clear; clc;
[AA, TXT, RAW]=XLSREAD('GW',1,'A2:F25');
to=size(RAW,1); i=1; flag=0; tipo=char(RAW(:,1));
corredor=char(RAW(2,2));

while i<=to & flag==0
    if strfind(tipo(i,:), 'out')
        flag=1; n_in=i-1;
    end
    m_out=to-i+1; i=i+1;
end

temp=RAW(:,6); temp1=char(temp);
IN(1:n_in,1)=str2num(temp1(1:n_in,1:2));
IN(1:n_in,2)=str2num(temp1(1:n_in,4:5));
IN(1:n_in,3)=str2num(temp1(1:n_in,7:8));
IN(1:n_in,4)=str2num(temp1(1:n_in,10));
OUT(1:m_out,1)=str2num(temp1(n_in+1:n_in+m_out,1:2));
OUT(1:m_out,2)=str2num(temp1(n_in+1:n_in+m_out,4:5));
OUT(1:m_out,3)=str2num(temp1(n_in+1:n_in+m_out,7:8));
OUT(1:m_out,4)=str2num(temp1(n_in+1:n_in+m_out,10));

j=0; k=0; in_par=[]; in_impar=[];
for i=1:n_in
    if rem(IN(i,2),2)==0
        j=j+1; in_par=[in_par,i];
        [DIST_PAR(j,:),r]=distcorr(IN(i,2:3),OUT(:,2:3));
        out_par=[r];
    else
        k=k+1; in_impar=[in_impar,i];
        [DIST_IMPAR(k,:),r]=distcorr(IN(i,2:3),OUT(:,2:3));
        out_impar=[r];
    end
end
end
```

```

[assignment,cost]=assignmentoptimal(DIST_IMPAR);
INImpares=in_impar'; OUTImpares=out_impar(assignment);
if n_in>m_out
    nesci=ones(size(INImpares));
    for i=1:size(assignment)
        nesci(assignment)=0;
    end
end
ORDEM1STR=[RAW(INImpares,6),RAW(n_in+OUTImpares,6)];

if size(DIST_IMPAR,1)>size(DIST_IMPAR,2)
    nesci=ones(size(in_impar)); nesci(assignment)=0;
    p=find(nesci==1); ORDEM1STR=[ORDEM1STR ;RAW(p,6),{' N.A.'}];
elseif size(DIST_IMPAR,1)<size(DIST_IMPAR,2)
    nescon=ones(size(out_impar)); nescon(assignment)=0;
    p=find(nescon==1);
    ORDEM1STR=[ORDEM1STR ;{' N.A.'}, RAW(n_in+p,6)];
end

[assignment,cost]=assignmentoptimal(DIST_PAR);
INpares=in_par'; OUTpares=out_par(assignment);
ORDEM2=[IN(INpares,:),OUT(OUTpares,:)];
ORDEM2STR=[RAW(INpares,6),RAW(n_in+OUTpares,6)];
if size(DIST_PAR,1)>size(DIST_PAR,2)
    nesci=ones(size(in_par)); nesci(assignment)=0;
    p=find(nesci==1);
    ORDEM2STR=[ORDEM2STR ;RAW(p,6),{' N.A.'}];
elseif size(DIST_PAR,1)<size(DIST_PAR,2)
    nescon=ones(size(out_par)); nescon(assignment)=0;
    p=find(nescon==1);
    ORDEM2STR=[ORDEM2STR ;{' N.A.'}, RAW(n_in+p,6)];
end

sprintf('Output for corridor %s',corredor)
disp('odd RACK')
ORDEM1STR
disp('even RACK')
ORDEM2STR
xlswrite('ORDER',ORDEM1STR,1)
xlswrite('ORDER',ORDEM2STR,2)

function [dstE_R,r]=distcorr(coord_ent,recolhas);
p_or_i=rem(coord_ent(:,1),2); [r]=find(rem(recolhas(:,1),2)==p_or_i);

```

```
n=size(r);
for j=1:n
    dstE_R(j)=sqrt((coord_ent(1)-recolhas(r(j),1))^2+...
                  (coord_ent(2)-recolhas(r(j),2))^2);
end
```

The "assignmentoptimal" function, returns an optimal assignment, in the sense of minimum cost using the Hungarian algorithm, and can be obtained from Matlab Central [12].