Chapter 20. Storage Systems

20.1. Unit Load Storage Policies

Introduction

Warehouse Operations Assumptions

Unit Load

The material in this chapter will focus on unit load warehouse operations. In unit load warehouses it is assumed that all the items in the warehouse are aggregated into units of the same size and can be moved, stored and controlled as a single entity. Typical examples of unit loads are pallets and wire baskets. It is also assumed that all the storage locations are the same size and each location can hold any unit load.

Figure 20.1. Unit Load Pallet Rack

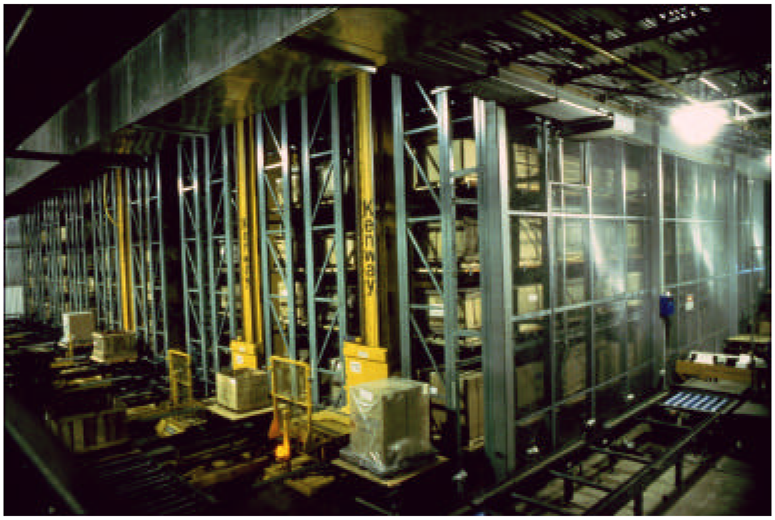


Figure 20.2. Unit Load Automated Storage/Retrieval System

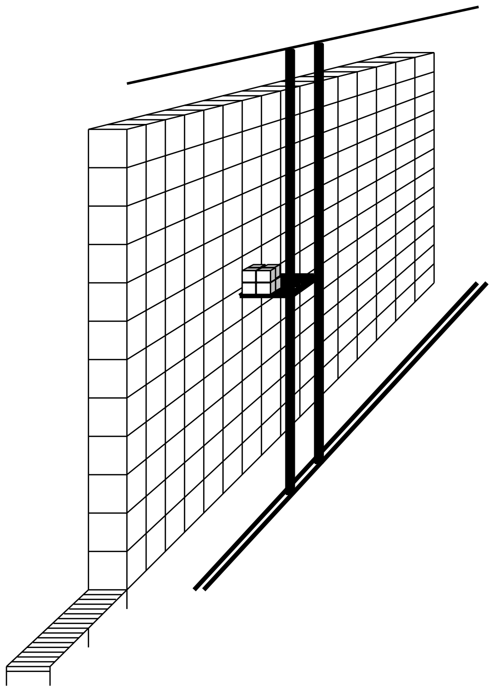


Figure 20.3. Unit Load AS/RS Illustration

Command Cycle

It is assumed that all the operations in the warehouse are performed in single command mode, i.e. the picker or crane performs a single operation on each round trip.

Travel Independence or Factoring Condition

If the travel independence or factoring condition is satisfied, then it is assumed that all the items in the warehouse have the same probability mass function for selection of a dock or input/output point. This allows the computation of the expected one way distance for each location, independent of which unit load will be stored in that location.

Main Warehousing Facilities Design Principle (Travel Time)

Place unit loads that generate the highest frequency of access in locations with the lowest expected distance.

Main Warehousing Facilities Design Principle (Storage Capacity)

Use the "Cube".

This principle encourages the use of the vertical dimension of the warehouse and the avoidance of empty unit locations. The vertical dimension of the warehouse can be used with block stacking storage systems and a large variety of rack storage systems. Empty unit storage locations can be avoided by the proper storage policy.

Shared versus Dedicated Storage Policies

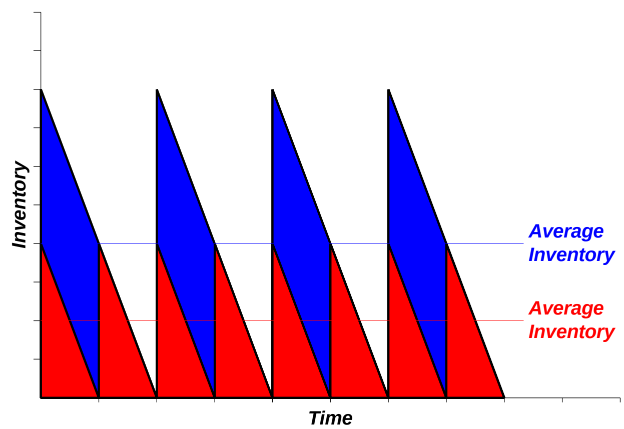


Figure 20.4. Cyclic Inventory Pattern

Dedicated Storage

With dedicated storage, a set of locations is reserved for the items of a single product during the entire planning period. The required warehouse size N is equal to the sum of the maximum inventories of each product. The location and management of items can be done by hand under relative stable demand conditions. Let I_{it} denote the inventory level of product *i* during period *t*, then

$$N_{DED} = \sum_i \left(\max_t I_{it} \right) \tag{20.1}$$

Shared Storage

With shared storage, a location can be used successively for the storage or items of different products. Examples of shared storage are random and closest open location storage. The required warehouse size N is equal to the maximum over time of the aggregate inventory. For many uncorrelated products, this size is half the size required by dedicated storage. Shared storage requires almost always a computerized system to manage and locate items in the warehouse, but this system has larger flexibility in adapting to changing demand conditions. Throughput comparisons depend on which shared storage policy is used, i.e. dedicated storage does not always minimize the expected travel time. This last statement is contrary to what is taught in many courses and is still controversial.

$$N_{SHA} = \max_t \{\sum_i I_{it}\} \tag{20.2}$$

$$a = \frac{N}{N_{DED}} \in [0.5, 1] \tag{20.3}$$

The ratio of the required warehouse size to the maximum required size under dedicated storage is called the sharing factor *a*. The sharing factor has a range of [0.5, 1]. This sharing factor can be most easily determined by simulation. Similarly, a warehouse balance *b* can be computed as

$$\begin{aligned} b &= 2(1-a) \\ a &= 1-\frac{b}{2} \end{aligned} \tag{20.4}$$

The sharing factor and warehouse balance indicate how well balanced the input and output flows of the warehouse are. A value of $a = 1$ or $b = 0$ indicate that the flows are not balanced at all. A value of $a = 0.5$ or $b = 1$ indicate that the flows are perfectly balanced.

Product Based Dedicated Storage

Consider the following warehouse layout. The warehouse has four rows of bays, with six bays in each row for a total of 24 bays. All the bays are 10 by 10 feet and only one product is stored per bay.

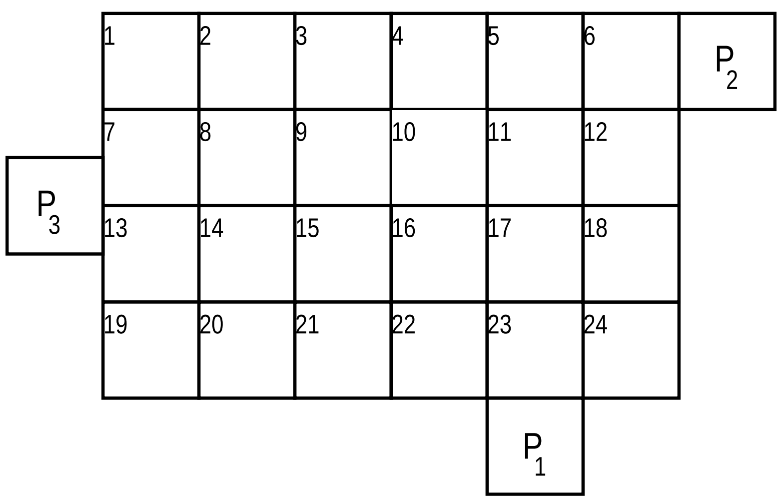


Figure 20.5. Warehouse Layout

Case 1: Factoring

All the material is received through the receiving door labeled P_3. Material is shipped through the two shipping doors labeled P_1 and P_2. All receipts and shipments are full pallet quantities. The following product information has been provided.

Table 20.1. Product Information (Factoring Case)

product	A	B	C
product storage requirements (q)	12	2	10
pallets received per month (r)	400	60	200
pallets shipped per month through			
door P1 (p1)	300	45	150
door P2 (p2)	100	15	50

The distance computations use a rectilinear distance norm between the centroid of each storage bay and the centroid of shipping/receiving areas. The warehouse operates under single command. A month is assumed to be 30 days.

We will first show the expected one-way travel distance for each location in the warehouse for the combined storage and retrieval of a single unit load. Then we will determine the best assignment of products to storage bays. Finally, we will compute the total travel per month for each product and for the total warehouse system.

Case 1 corresponds to a dedicated storage policy with the travel independence or factoring condition satisfied. It can be solved by hand by the innerproduct minimization of the frequency of access and expected travel time vectors by sorting them in opposite directions.

The probability mass functions for the three products are all equal to (0.375, 0.125, 0.500), hence the travel independence conditions is satisfied.

The expected travel one-way travel distance for location 1 is then given by:

$$D_j = \sum_{k=1}^{K} p_k d_{kj} \tag{20.5}$$

For example, g_1=0.375·80+0.125·60+0.500·25=50.0.

If a single unit square is moved the delta is 3.75, 1.25, and 5.00, with respect to the first, second, and third dock, respectively. Moving from square 1 to square 2 to gives a delta of -3.75-1.25+5.00=0. Moving from square 5 to square 6 gives a delta of 3.75-1.25+5.00=7.5. Moving from square 1 to square 7 gives a delta of -3.75+1.25-0.50=-7.5. Moving from square 7 to square 13 gives a delta of -3.75+1.25=-2.5. Finally, moving from square 7 to square 19 gives a delta of -3.75+1.25+5.00=2.5. The resulting expected travel distances are shown at the bottom of the unit squares in Figure 20.6.

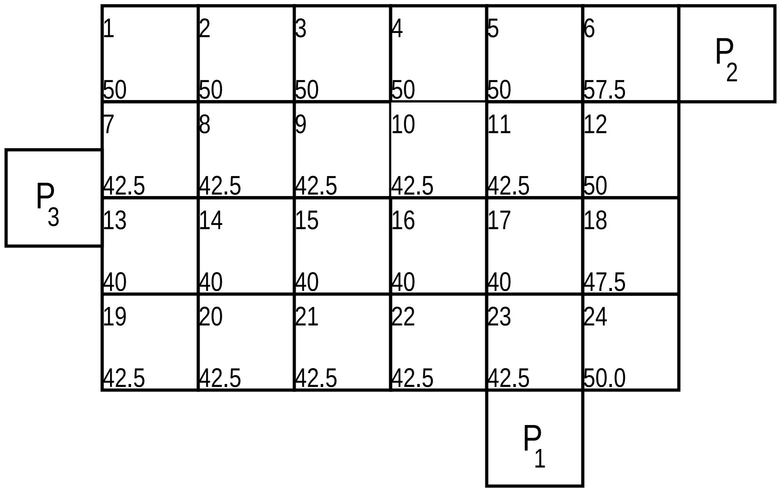


Figure 20.6. Expected one-way travel distances for all products.

Product Turnover-Based

To find the order in which to locate products, the "frequency-of-access" of each product must be computed as the ratio of monthly demand divided by number of bays, i.e.

$$f_A = \frac{r_A}{q_A} \tag{20.6}$$

This is similar to the "cube-per-order-index" introduced by Heskett (1963,1964). The frequencies of access for the products are:

$f_A = 400/12 = 33.33$
$f_B = 60/2 = 30$
$f_C = 200/10 = 20$

The products are assigned by decreasing frequency of access to the locations by increasing expected travel time. This is equivalent to minimizing the innerproduct of two vectors by sorting them in opposite directions. Hence product A get assigned first, then product B, and finally product C. The assignments are shown in the Figure 20.7.

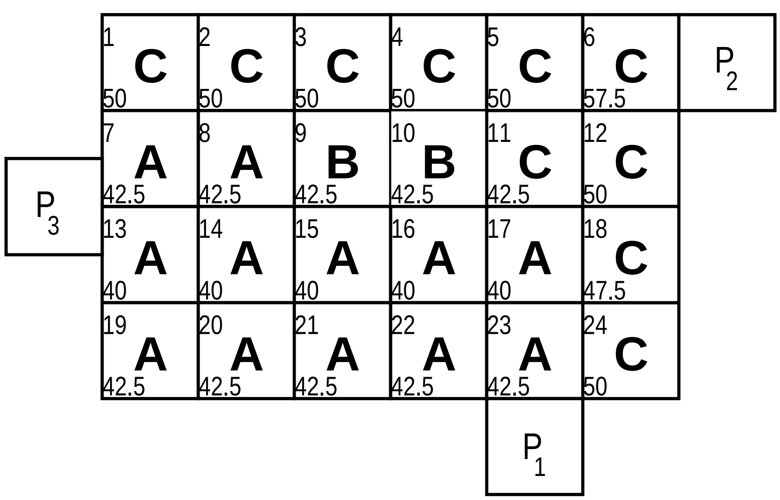


Figure 20.7. Assignment Solution for the Factoring Case

The total travel cost per month (time period) is computed first by product and then summed over all products. Let Z_A be the set of locations associated with product A and let t_A be the average one way travel distance to a location assigned to product *A*, then the total travel for product *A* is given by:

$$T_A = 4r_A t_A = 4r_A \left(\frac{\sum_{j \in \mathbf{Z}_A} D_j}{q_A} \right) = 4 f_A \left(\sum_{j \in \mathbf{Z}_A} D_j \right) \tag{20.7}$$

$$T = \sum_{p=1}^{P} T_p \tag{20.8}$$

The total travel distances per month for the products are then:

T_A = 4·400·(5·40+7·42.5)/12 = 4·400·497.5/12 = 4·400·41.46 = 66,333
T_B = 4·60·(2·42.5)/2 = 4·60·42.5 = 10,200
T_C = 4·200·(42.5+47.5+7·50+57.5)/10 = 4·200·497.5/10 = 4·200·49.75 = 39,800
T = T_A + T_B + T_C = 66,333 + 10,200 + 39,800 = 116,333

Two commonly used storage policies are based on sequencing the products by decreasing operations or by increasing required storage space. This first policy is commonly denoted by "putting the fast movers closest to the door." The second policy could be referred to as "putting the low inventory products closest to the door." The average montly travel time for each policy will be computed for the above example.

Demand Based

The "fast movers" are identified by the largest demand or, equivalently, by the largest number of operations. The products are then sorted by decreasing number of operations. The number of operations for the three products are:

$f_A = 400,\ f_C = 200,\ f_B = 60$

The products are assigned by decreasing number of operations to the locations by increasing expected travel time. This is equivalent to minimizing the innerproduct of two vectors by sorting them in opposite directions. Hence product A get assigned first, then product C, and finally product B. The assignments are shown in the Figure 20.8.

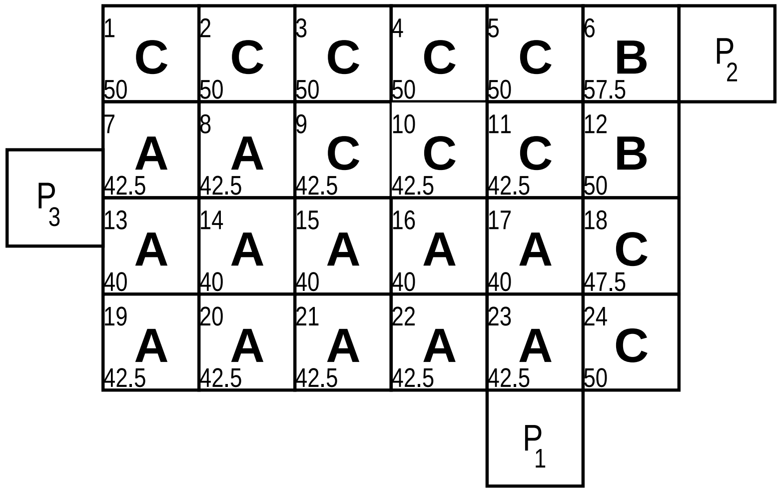


Figure 20.8. Warehouse Layout for the Factoring Case based on Demand

The total travel cost per month (time period) is computed first by product and then summed over all products.

The total travel distances per month for the products are then:

TA = 4·400·(5·40+7·42.5)/12 = 4·400·497.5/12 = 4·400·41.46 = 66,333
TB = 4·60·(50+57.5)/2 = 4·60·107.5/2= 4·60·53.75 = 12,900
TC = 4·200·(3·42.5+47.5+6·50)/10 = 4·200·475.0/10 = 4·200·47.50 = 38,000
T = TA + TB + TC = 66,333 + 12,900 + 38,000 = 117,233

This is an increase of 0.8 % over the optimal turnover-based dedicated storage policy.

Storage Size Based

The "low inventory" or “small inventory” products are identified by their required number of storage locations. The products are then sorted by increasing number of storage locations. The number of storage locations for the three products are:

$$f_A = 12,\ f_C = 10,\ f_B = 2$$

The products are assigned by decreasing number of operations to the locations by increasing expected travel time. This is equivalent to minimizing the innerproduct of two vectors by sorting them in opposite directions. Hence product B get assigned first, then product C, and finally product A. The assignments are shown in the Figure 20.9.

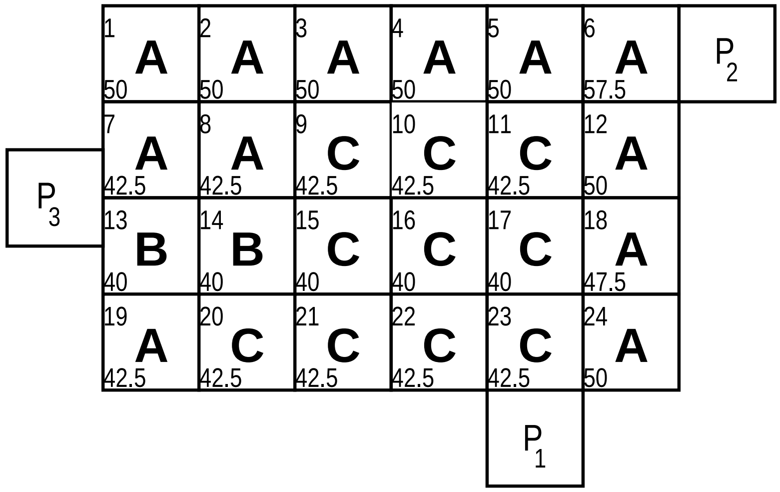


Figure 20.9. Warehouse Layout for the Factoring Case based on Storage Locations

The total travel cost per month (time period) is computed first by product and then summed over all products.

The total travel distances per month for the products are then:

$TA = 4 \cdot 400 \cdot (3 \cdot 42.5 + 47.5 + 7 \cdot 50 + 57.5)/12 = 4 \cdot 400 \cdot 582.5/12 = 4 \cdot 400 \cdot 48.54 = 77{,}667$
$TB = 4 \cdot 60 \cdot (2 \cdot 40)/2 = 4 \cdot 60 \cdot 80/2 = 4 \cdot 60 \cdot 40 = 9{,}600$
$TC = 4 \cdot 200 \cdot (3 \cdot 40 + 7 \cdot 42.5)/10 = 4 \cdot 200 \cdot 417.5/10 = 4 \cdot 200 \cdot 41.75 = 33{,}400$
$T = TA + TB + TC = 77{,}667 + 9{,}600 + 33{,}400 = 120{,}667$

This is an increase of 3.7 % % over the optimal turnover-based dedicated storage policy.

Case 2: Non-Factoring

Consider the following shipping pattern, with the same storage requirements as given before.

Table 20.2. Product Information (Non-Factoring Case)

Product	A	B	C
Product storage requirements (q)	12	2	10
Pallets received and shipped per month through			
door P_1	300	6	100
door P_3	100	24	240
door P_2	400	90	60

Case 2 corresponds to a dedicated storage policy where the travel independence or factoring condition is not satisfied. It must be solved with a transportation or assignment model.

The probability mass functions for the three products are given in the following table:

Table 20.3. Probability mass functions in percent (Case 2).

product	A	B	C
door P1 (p_1)	37.5	5	25
door P2 (p_2)	12.5	20	60
door P3 (p_3)	50	75	15

The probability mass function for product A is the same as in case 1, and hence the expected one way travel distances for product A are given in Figure 20.2. The expected one way travel distances for products B and C are computed in a similar manner and are given in Figures 20.10 and 20.11, respectively.

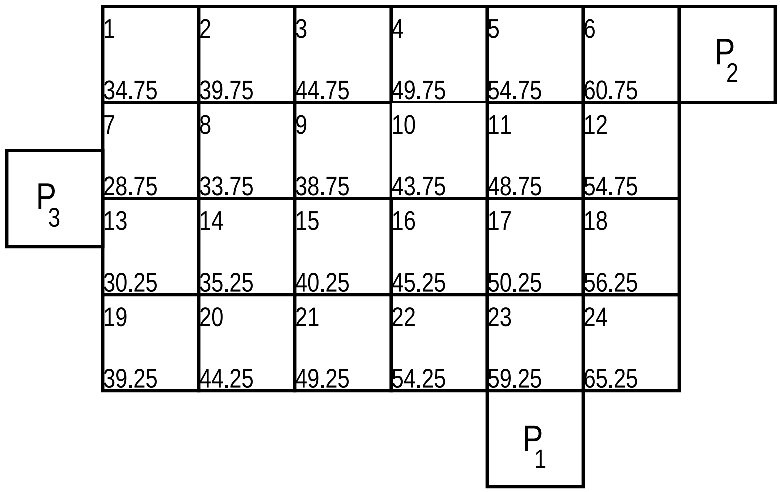


Figure 20.10. Expected One-way Travel Distances for Product B

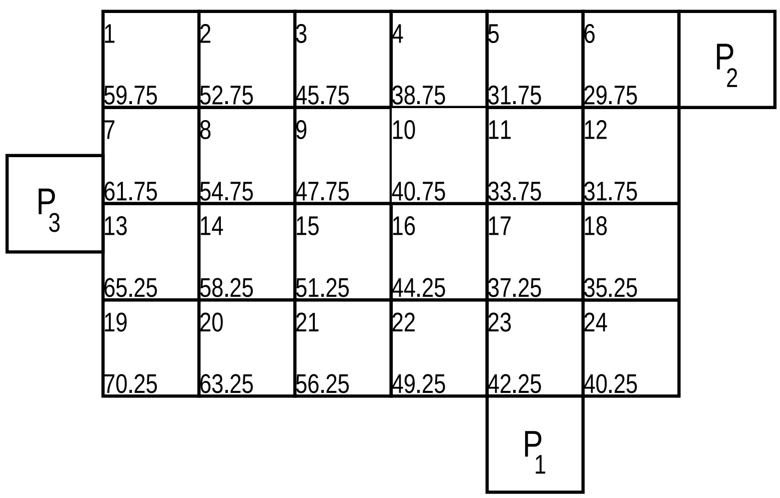


Figure 20.11. Expected One-way Travel Distances for Product C

The linear transportation formulation is then:

$$
\begin{aligned}
&Min \quad && \sum_{i=1}^{M}\sum_{j=1}^{N} f_i D_{ij} x_{ij} \\
&s.t. && \sum_{j=1}^{N} x_{ij} = q_i \\
& && \sum_{i=1}^{M} x_{ij} \le 1 \\
& && x_{ij} \ge 0
\end{aligned}
\tag{20.9}
$$

A linear programming model, compatible with the Take command of LINDO or the Read LP format of CPLEX, is given next. The results of the linear programming model are shown in Figure 20.12.

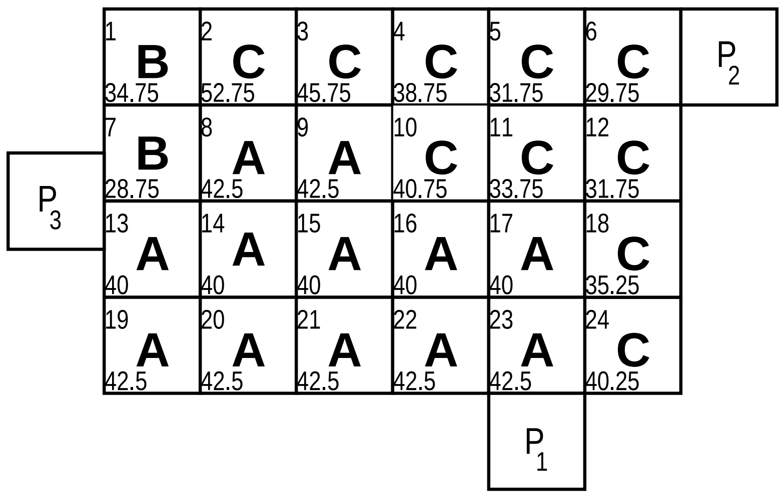


Figure 20.12. Assignment Solution for the Non-Factoring Case

The total travel times for the products are then:

$T_A = 4 \cdot 400 \cdot 497.5/12 = 66{,}333.33$
$T_B = 4 \cdot 60 \cdot 63.5/2 = 7620$
$T_C = 4 \cdot 200 \cdot 380.50/10 = 4 \cdot 200 \cdot 38.05 = 30{,}440$
$T = T_A + T_B + T_C = 66{,}333.33 + 7{,}620 + 30{,}440 = 104{,}393.33$

The factoring storage policy for the same three products with the same total demand but with different material handling moves requires 11.44 % more travel time than the non-factoring policy.

Linear Programming Model for Non-Factoring Case

Let XIJ be equal to one if the product I is assigned to location J. Let SGI be the sum of the one way travel distances to all locations assigned to product I. The objective is to minimize innerproduct of the frequence of access of each product times the total travel distance for all locations assigned to that product. The first three constraints are the definition of the SGI's for each product. The next three constraints ensure that there are enough locations assigned to each product. Finally, the last constraints ensure that each location holds at most one unit load.

Code Listing 1. Non-Factoring Storage Policy LP Formulation

```
MIN 133.3333 SGA + 120 SGB + 80 SGC
SUBJECT TO
50 XA1 + 50 XA2 + 50 XA3 + 50 XA4 + 50 XA5 + 57.5 XA6 +
```

```
42.5 XA7 + 42.5 XA8 + 42.5 XA9 + 42.5 XA10 + 42.5 XA11 + 50 XA12 +
40 XA13 + 40 XA14 + 40 XA15 + 40 XA16 + 40 XA17 + 47.5 XA18 +
42.5 XA19 + 42.5 XA20 + 42.5 XA21 + 42.5 XA22 + 42.5 XA23 + 50 XA24
- SGA = 0
34.75 XB1 + 39.75 XB2 + 44.74 XB3 + 49.75 XB4 + 54.75 XB5 + 60.75 XB6 +
28.75 XB7 + 33.75 XB8 + 38.75 XB9 + 43.75 XB10 + 48.75 XB11 +
54.75 XB12 + 30.25 XB13 + 35.25 XB14 + 40.25 XB15 + 45.25 XB16 +
50.25 XB17 + 56.25 XB18 + 39.25 XB19 + 44.25 XB20 + 49.25 XB21 +
54.25 XB22 + 59.25 XB23 + 65.25 XB24
- SGB = 0
59.75 XC1 + 52.75 XC2 + 45.75 XC3 + 38.75 XC4 + 31.75 XC5 + 29.75 XC6 +
61.75 XC7 + 54.75 XC8 + 47.75 XC9 + 40.75 XC10 + 33.75 XC11 +
31.75 XC12 + 65.25 XC13 + 58.25 XC14 + 51.25 XC15 + 44.25 XC16 +
37.25 XC17 + 35.25 XC18 + 70.25 XC19 + 63.25 XC20 + 56.25 XC21 +
49.25 XC22 + 42.25 XC23 + 40.25 XC24
- SGC = 0
XA1 + XA2 + XA3 + XA4 + XA5 + XA6 + XA7 + XA8 + XA9 + XA10 +
XA11 + XA13 + XA14 + XA15 + XA16 + XA17 + XA18 + XA19 + XA20 + XA21 +
XA22 + XA23 + XA24 + XA12 =    12
XB1 + XB2 + XB3 + XB4 + XB5 + XB6 + XB7 + XB8 + XB9 + XB10 +
XB11 + XB13 + XB14 + XB15 + XB16 + XB17 + XB18 + XB19 + XB20 + XB21 +
XB22 + XB23 + XB24 + XB12 =    2
XC1 + XC2 + XC3 + XC4 + XC5 + XC6 + XC7 + XC8 + XC9 + XC10 +
XC11 + XC13 + XC14 + XC15 + XC16 + XC17 + XC18 + XC19 + XC20 + XC21 +
XC22 + XC23 + XC24 + XC12 =    10
XA1 + XB1 + XC1 <= 1
XA2 + XB2 + XC2 <= 1
XA3 + XB3 + XC3 <= 1
XA4 + XB4 + XC4 <= 1
XA5 + XB5 + XC5 <= 1
XA6 + XB6 + XC6 <= 1
XA7 + XB7 + XC7 <= 1
XA8 + XB8 + XC8 <= 1
XA9 + XB9 + XC9 <= 1
XA10 + XB10 + XC10 <= 1
XA11 + XB11 + XC11 <= 1
XA12 + XB12 + XC12 <= 1
XA13 + XB13 + XC13 <= 1
XA14 + XB14 + XC14 <= 1
XA15 + XB15 + XC15 <= 1
XA16 + XB16 + XC16 <= 1
XA17 + XB17 + XC17 <= 1
XA18 + XB18 + XC18 <= 1
XA19 + XB19 + XC19 <= 1
XA20 + XB20 + XC20 <= 1
XA21 + XB21 + XC21 <= 1
XA22 + XB22 + XC22 <= 1
XA23 + XB23 + XC23 <= 1
XA24 + XB24 + XC24 <= 1
END
```

Product Turnover Class Based Storage

Pure Dedicated is Very Space Inefficient

3 to 5 Classes based on Frequency of Access

Dedicated Space for Each Class

Class Space Determined by Simulation

Inside Class Use Random or Closest Open Location

Shared Storage Policy

Duration-of-Stay Shared Storage

Illustration

4 Products (A, B, C, D)

Replenishment Batch Size q = 4 Unit Loads

Demand Rate r = 1 Unit Load / Day

Replenishment Days (A = 1, B = 2, C = 3, D = 4)

AS/RS Storage ($v_x = v_y = 1$)

	3	3	3	4
	2	2	3	4
	1	2	3	4
I/O	1	2	3	4

Figure 20.13. Storage Example Rack Travel Times

	3 D	3 D	3 D	4 D
	2 C	2 C	3 C	4 C
	1 B	2 B	3 B	4 B
I/O	1 A	2 A	3 A	4 A

Figure 2014. Product Dedicated Optimal Storage

Each Product Turnover Rate = 1/4

Any Storage Assignment is Optimal

Maximum Storage Space = 16

Average Daily Travel = (10 + 10 + 11 + 13) / 4 = 11

	3 A_4	3	3	4
	2 A_3	2 C_3	3 B_4	4
	1 A_2	2 D_3	3 C_4	4
I/O	1 A_1	2 D_2	3 D_4	4

	3 A_4	3	3	4
	2 A_3	2 B_3	3 B_4	4
	1 A_2	2 D_3	3 C_4	4
I/O	1 B_1	2 B_2	3 D_4	4

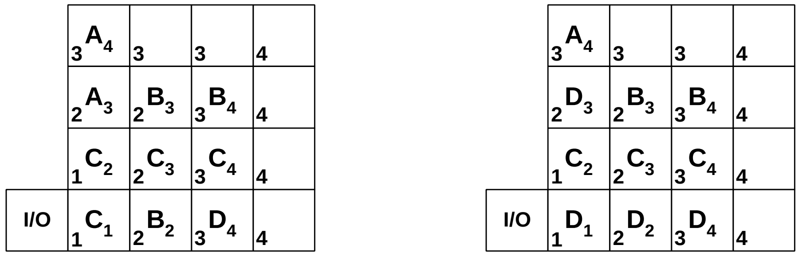


Figure 2015. Duration-Of-Stay Storage Patterns On the Different Days

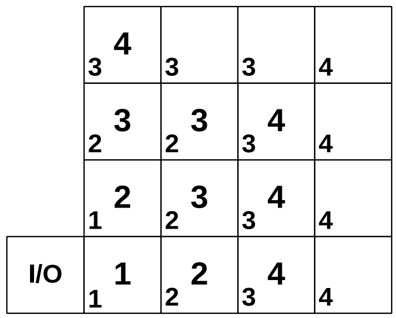


Figure 20.16. Duration-Of-Stay Optimal Storage

Store by Increasing “Duration-Of-Stay” (DOS)

Minimum Storage Space = 10

Average Daily Travel = (1 + 3/2 + 6/3 + 12/4) = 7.5

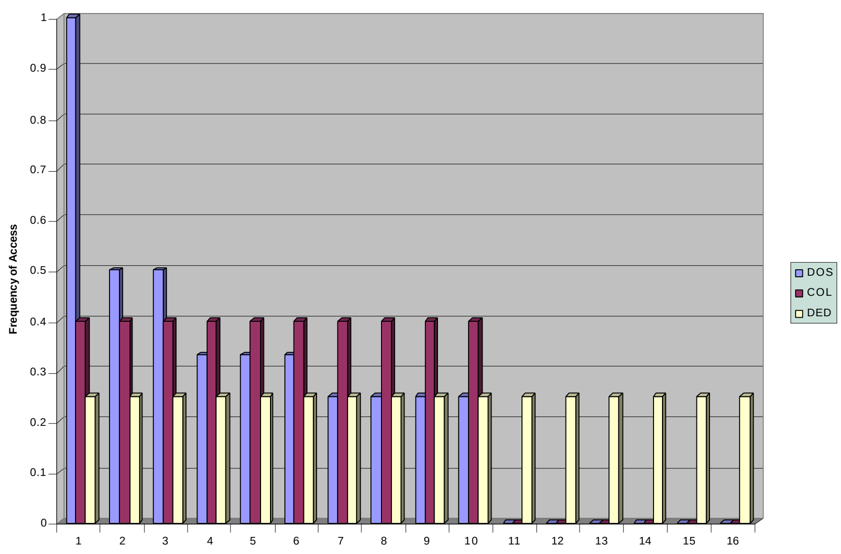


Figure 20.17. Frequency of Access Distribution for Various Storage Policies

Required Storage Space: 10 Shared, 16 Dedicated, + 60 %

Expected Travel Time: 7.5 Shared, 11 Dedicated, + 47 %

Accesses to Best Location: 1 Shared, 0.25 Dedicated, -75 %

Exploits that First and Last Unit Load in Batch are Different

Cross Docking (DOS = 0)

Minimizes Both Storage Space and Travel Time for a Perfectly Balanced Warehouse

Very Constrained Perfectly Balanced Replenishment Pattern $n_p(t)$

Perfectly Balanced Warehouse

Balanced = Minimum Space

Perfectly Balanced = Minimum Space and Minimum Time

$$n_p(t) = n_p(t+p) \qquad \forall t, \forall p$$
$$z_p = \sum_{i=1}^{p} n_p(i) \tag{20.10}$$

Example

Consider the following product data and the same warehouse layout. All the material is received through the receiving door labeled P_3. Material is shipped through the two shipping doors labeled P_1 and P_2. For each product three times as much material is shipped through door P_1 as through door P_2. All receipts and shipments are full pallet quantities. The warehouse operates under single command. The travel time is measured centroid to centroid with the rectilinear distance norm. All the bays are 10 by 10 feet and only one item is stored per bay. The best shared storage policy is used. Products are replenished during the day their inventory reaches zero. The expected one-way travel distance for each location in the warehouse for the combined storage and retrieval of a single unit load is computed as shown in Figure 20.618

Table 20.4. Product Information (Example 2).

product	daily demand	reorder quantity	replenishment day
A	1	4	3
B	0.25	2	2
C	1	4	2
D	1	4	1
E	0.25	3	3
F	0.25	3	7
G	1	4	4
H	0.25	2	6
I	0.25	3	11

There are three groups of products that have the same daily demand and reorder quantity. The groups consists of products A, C, D, and G (group 1), products B and H (group 2), and products E, F, and I (group 3). First we check if each of the groups satisfies the perfectly balanced condition. For instance, for group 2 on day 2 a unit load with duration of stay 4 and 8 is withdrawn and product B is replenished which deposits a unit load with duration of stay 4 and 8 days. So group 2 is perfectly balanced. Similar computations show that each group does satisfy the perfectly balance condition. Next we construct the input/output diagram by duration of stay shown in Table 20.5.

Table 20.5. Input/Output Diagram by Duration of Stay

	1	2	3	4	5	6	7	8	9	10	11	12	zone
1	D	C	A	G									1
2	D	C	A	G									2
3	D	C	A	G									3
4	D	C, B	A, E	G		H	F				I		6
5													0
6													0
7													0
8		B	E			H	F				I		4
9													0
10													0
11													0
12			E				F				I		3

The last column shows how large the zones have to be for each of the durations of stay. Only the unit loads in the first p days have to be summed to find the zone size for loads of duration of stay p, since at period p+1 the pattern repeats itself. The relevant unit loads are shown in bold in Table 20.5 to the left of and below the staircase line. The resulting warehouse layout is shown in Figure 19. The number in each bay indicates the duration of stay of any load stored in this bay, no longer the product label. The travel times are then computed first by duration of stay zone and then for the whole warehouse. The total number of slots used is equal to 19, even though the total number of slots required for dedicated storage would have been 29. Notice that not all the storage bays are used when using the shared storage policy. In fact, for the perfectly balanced case, the number of storage bays used is the smallest possible. In addition, the average travel is also minimized.

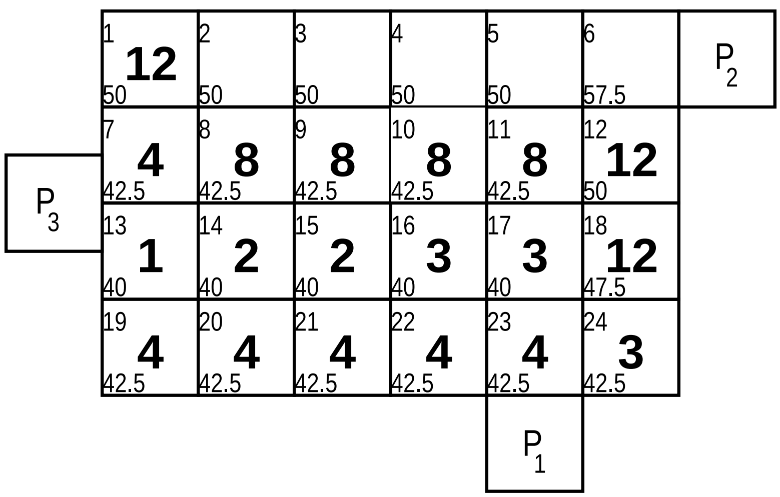


Figure 20.19 Duration of Stay Warehouse Zones

The travel distance per duration of stay (DOS) zone is then computed with the following formula, where z_{DOS} is the size of the duration of stay zone Z_{DOS} and t_{DOS} is the average one-way travel distance to a location in this zone.

$$T_{DOS} = 4\frac{1}{DOS} z_{DOS} t_{DOS} = \frac{4}{DOS} z_{DOS} \left(\frac{\sum_{j \in \mathbf{Z}_{DOS}} D_j}{z_{DOS}} \right) = \frac{4}{DOS} \left(\sum_{j \in \mathbf{Z}_{DOS}} D_j \right) \qquad (20.11)$$

$$T = \sum_{DOS} T_{DOS} \qquad (20.12)$$

$$T_1 = 4 \cdot 40 / 1 = 160$$
$$T_2 = 4 \cdot (40 + 40) / 2 = 160$$
$$T_3 = 4 \cdot (40 + 40 + 42.5) / 3 = 163.33$$
$$T_4 = 4 \cdot (6 \cdot 42.5) / 4 = 255$$
$$T_8 = 4 \cdot (3 \cdot 42.5 + 47.5) / 8 = 87.5$$
$$T_{12} = 4 \cdot (3 \cdot 50) / 12 = 50$$
$$T = T_1 + T_2 + T_3 + T_4 + T_8 + T_{12} = 875.83$$

Further information can be found in Goetschalckx and Ratliff (1990).

Not Perfectly Balanced Warehousing Systems

Static Greedy Heuristic

- Sort by Increasing Departure Time

Adaptive, Dynamic Heuristic

- Combine DOS into classes
- Remedial Action for Full Classes

$$z_p = p \cdot E\left[n_p\right] \qquad (20.13)$$

Closest-Open-Location and Random Shared Storage

For comparison purposes we can also compute the expected travel distance under closest open location storage policy. The closest open location storage policy is equivalent to the pure random storage policy if all locations in the rack are used. For our example, the required number of locations is the same as under perfectly balanced shared storage, since the input and output flows are identical. The 19 locations with lowest travel distance will be used and the average travel distance will be based only on those 19 locations.

$$T_i = 4 f_i q_i \left(\frac{\sum_{j=1}^{N} d_j}{N} \right) = 4 f_i q_i \bar{d} = 4 r_i \bar{d} \qquad (20.14)$$

$$T_{RAN} = \sum_i T_i = 4\bar{d} \sum_i r_i \qquad (20.15)$$

d = (5 · 40 + 10 · 42.5 + 47.5 + 3 · 50) / 19 = 822.50 / 19 = 43.29
T = 4 · 43.29 · (4 · 1 + 5 · 0.25) = 909.08

Comparison of Storage Policies

Comparison Example

Given a warehouse configuration as shown in the next Figure with a total of 18 locations and three docks. Each location and each dock is assumed to be 10 feet wide by 10 feet long. The travel is assumed to be rectilinear from location centroid to location centroid and the warehouse is assumed to operate under single command. It is assumed that the loads can travel through the dock areas if required. The product data are given in the following Table. All products depart through dock P3, 20 % of all products arrive through dock P1, 20 % of all products arrive through dock P2, and 60 % of all products arrive through dock P3.

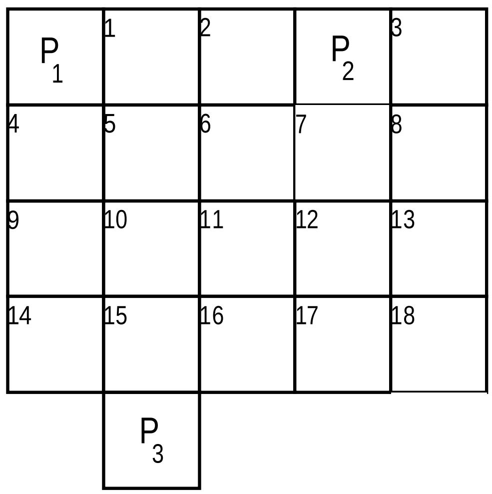


Figure 20.20. Warehouse Layout for Comparison of Storage Policies

Table 20.6. Comparison of Storage Policies Product Data

Product	Demand Rate	Reorder Quantity	Re supply Period
A	0.5	2	2
B	1	2	2
C	1	3	3
D	1	2	1
E	0.5	2	4
F	1	3	2
G	1	3	1

The solution procedure will execute a sequence of computations to arrive at the optimal warehouse layout under the various storage policies and single command and will finally compare the performance of the policies.

First, the dock selection probability mass functions are computed for each product and the travel independence condition is verified. Since all products have the same interface pattern with the docks the travel independence condition is satisfied and there is only one probability mass function. Its values are:

$p_1 = (0.2 + 0) / 2 = 0.1$
$p_2 = (0.2 + 0) / 2 = 0.1$
$p_3 = (0.6 + 1.0) / 2 = 0.8$

Second, the expected round trip storage and retrieval travel distances for each location are computed.

$g_1 = 4 \cdot (0.1 \cdot 10 + 0.1 \cdot 20 + 0.8 \cdot 40) = 4 \cdot 35 = 140$
$g_2 = 4 \cdot (0.1 \cdot 20 + 0.1 \cdot 10 + 0.8 \cdot 50) = 4 \cdot 43 = 172$
$g_3 = 4 \cdot (0.1 \cdot 40 + 0.1 \cdot 10 + 0.8 \cdot 70) = 4 \cdot 61 = 244$
$g_4 = 4 \cdot (0.1 \cdot 10 + 0.1 \cdot 40 + 0.8 \cdot 40) = 4 \cdot 37 = 148$
$g_7 = 4 \cdot (0.1 \cdot 40 + 0.1 \cdot 10 + 0.8 \cdot 50) = 4 \cdot 45 = 180$

The difference in total distance when moving one location "down" is $4 \cdot (1 + 1 - 8) = 4 \cdot (-6) = -24$. This allows the easy computation of the total distances for the rest of the locations. The results are shown in the next Figure.

P_1	1 140	2 172	P_2	3 244
4 148	5 116	6 148	7 180	8 220
9 124	10 92	11 124	12 156	13 196
14 100	15 68	16 100	17 132	18 172
	P_3			

Figure 20.21. Expected Storage and Retrieval Round Trip Travel Distances

Third, the frequency of access for each product under product turnover dedicated storage is computed and the products are ranked by non-increasing frequency of access. Then the locations are assigned to the products based upon this ranking. If there are ties in the selection of locations, then products are kept together as much as possible. Breaking these ties will lead to alternative warehouse layouts that have the same overall distance score.

Table 20.7. Frequency of Access Computation and Rank

Product	Demand Rate	Reorder Quantity	Re supply Period	Frequency of Access	Rank
A	0.5	2	2	0.25	3
B	1	2	2	0.5	1
C	1	3	3	0.33	2
D	1	2	1	0.5	1
E	0.5	2	4	0.25	3
F	1	3	2	0.33	2
G	1	3	1	0.33	2

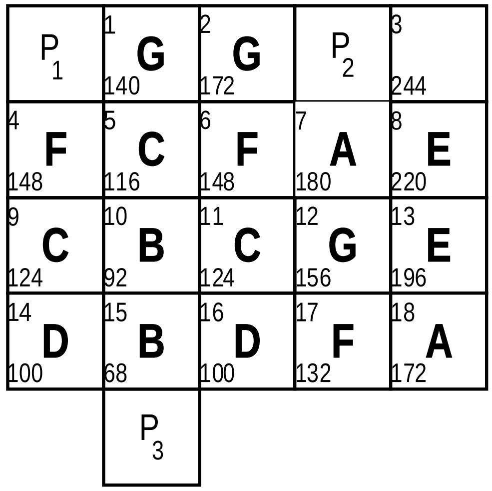


Figure 20.22. Product Storage Layout

Fourth, the total travel distance per products is computed and then these total travel distances are added to yield the overall travel for the product turnover dedicated warehouse layout. Observe also that the required warehouse size for dedicated storage is equal to the sum of the reorder quantities, which is 17 in this example.

$T_A = 0.25 \cdot (172 + 180) = 88$
$T_B = 0.5 \cdot (68 + 92) = 80$
$T_C = 0.33 \cdot (124 + 124 + 116) = 121.33$
$T_D = 0.5 \cdot (100 + 100) = 100$
$T_E = 0.25 \cdot (196 + 220) = 104$
$T_F = 0.33 \cdot (132 + 148 + 148) = 142.67$
$T_G = 0.33 \cdot (140 + 156 + 172) = 156$
$T = 88 + 80 + 121.33 + 100 + 104 + 142.67 + 156 = 792$

Fifth, we verify that each group of products is perfectly balanced. Then we construct the table showing the number of unit loads of each product with their arrival period and duration of stay. Summing the first p days for each duration of stay p then yields the required zone size (i.e. number of locations) for that duration of stay. The results are given in the following table. The total required warehouse size is equal to the sum of the zone size, which is equal to 12 in this example.

Table 20.8. Input/Output Diagram by Duration of Stay

	1	2	3	4	zone
1	D,G	B,F	C		2
2	D,G	A,B,F	C	E	5
3	G	F	C		3
4		A		E	2

Sixth, the optimal layout for duration of stay storage policies is determined by assigning the zones with the smallest duration of stay to the locations with the lowest expected travel distances. The results are shown in the next Figure.

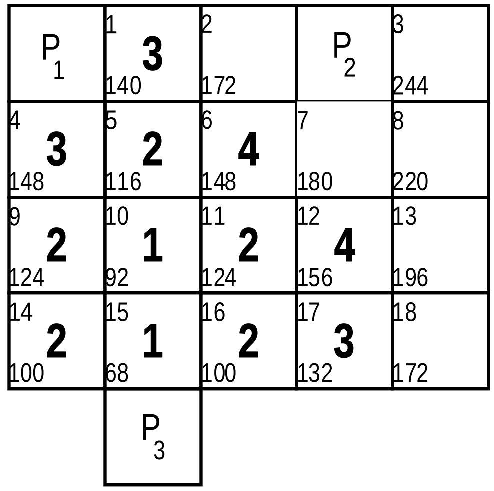


Figure 20.23. Unit Duration-Of-Stay Warehouse Layout

Seventh, the total travel distance for the duration of stay storage policy is computed by first computing the total travel distance per duration of stay zone and than adding all these travel distances together.

$T_1 = (1/1)\cdot(68 + 92) = 160$
$T_2 = (1/2)\cdot(100 + 100 + 124 + 124 + 116) = 282$
$T_3 = (1/3)\cdot(148 + 140 + 132) = 140$
$T_4 = (1/4)\cdot(148 + 156) = 76$
$T = 160 + 282 + 140 + 76 = 658$

Eight and last, the space and travel distance ratios for the product-turnover dedicated storage and the duration-of-stay shared storage are computed.

$$\frac{T_{DOS}}{T_{DED}} = \frac{658}{792} = 83\%$$

$$\frac{N_{DOS}}{N_{DED}} = \frac{12}{17} = 70\%$$

Comparison of Storage Policies Experiment

Policies: DOS, COL, DED, 2CL, 2 ZN

Products: 10, 20, 40, 80

Batch Size: 5, 10, 20, 40

3 Replications

Random First Replenishment Period

Simulation of Deterministic System

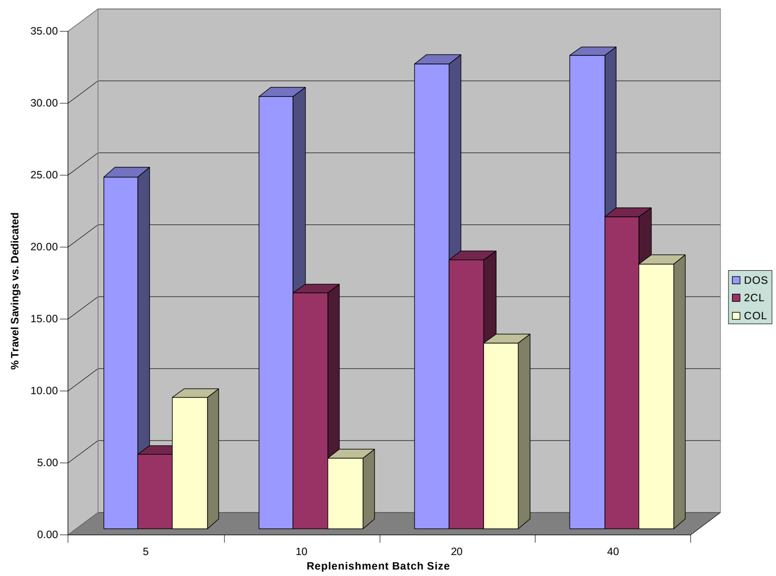


Figure 20.24. Influence of the Batch Size on the Performance of Storage Policies

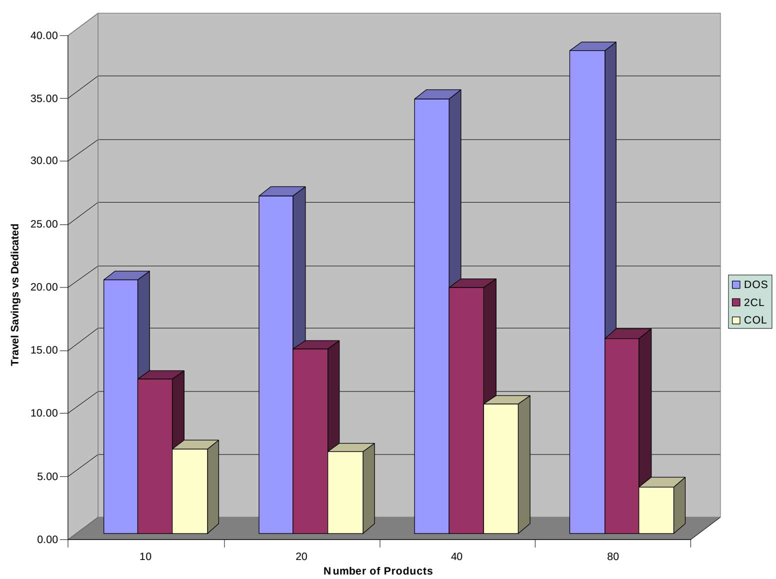


Figure 20.25. Influence of the Number of Products on the Performance of Storage Policies

Experimental Comparison Summary

Adaptive DOS is Superior (20 - 30 % Savings)

Two Class is Next Best (15 % Savings)

Two Zone is not as good as expected

Full Turnover Dedicated is Worst of All

Comparison of Storage Policies Exercise

Four products are stored in the warehouse shown in Figure 20.26. Assume rectilinear distance between the dock centroid, indicated by the circle in the Figure 20.26, and the centroid of the storage bays. Furthermore, assume single command travel cycles. Each storage bay measures 20 by 20 feet. Eighty storage bays are available for storage. A product is replenished when its inventory reaches zero, i.e., there is no safety stock. The replenishment quantities, the number of demand operations per day, and the arrival day for each product are given in the Table 20.9. The warehouse is operating as a stationary cyclical process. The arrival day is the day in the cycle that the product gets replenished.

Table 20.9. Product Information

Product	Replenishment Quantity	Demand per Day	Arrival Day
A	12	4	1
B	28	7	1
C	24	4	3
D	16	4	2

Determine first if the travel independence condition is satisfied. Then determine the expected one way travel distance to each of the bays.

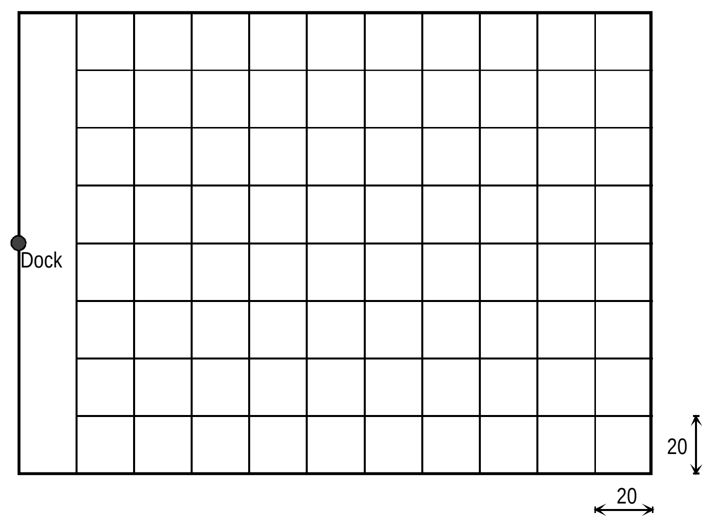


Figure 20.26. Warehouse Layout

Determine the optimal product dedicated storage layout that minimizes the expected travel distance per day. Compute the expected travel time per product and the total expected travel time for this warehouse layout. Assume that the products are stored by decreasing demand in the most desirable locations. Compute the expected travel time per product and the total expected travel time for this warehouse layout. Assume that the products are stored by increasing required storage in the most desirable locations. Compute the expected travel time per product and the total expected travel time for this warehouse layout. What are the required space penalty and travel time penalty (i.e. excess over the best policy) for each policy for this case. Summarize your answer in a clear table.

Assume that the arrival day for products B and C are swapped, i.e. product B now arrives on day 3 and product C now arrives on day 1 of the warehouse cycle. What are the required space penalty and travel time penalty (i.e. excess over the best policy) for each policy for this case. Summarize your answer in a clear table.

Suppose that random storage, rather than product dedicated storage, is used in this warehouse. Assume that replenishments for a day occur after all the demands for that day have been satisfied. What is the cycle for this warehouse operating under random storage policy? What is the number of units present in the warehouse at the end of each day of the cycle? What is the maximum number of storage bays required for storing the products using random storage? Show the warehouse layout for random storage at the end of day five of the cycle. Compute the expected travel time per day for every day in the cycle and the total expected travel time for this warehouse layout. What is the warehouse size, the sharing factor, and the balance of this warehouse system for the random storage policy? Discuss the advantages and disadvantages of random storage versus product dedicated storage.

This problem has been adapted from Tompkins and White (1984). The total expected travel time for product dedicated storage is 11,200. The required warehouse size for random storage is 69 and the total expected travel time is 8,897.

Storage Policy Comparison

Three products are stored in the warehouse shown in Figure 20.26. Products arrive at the receiving dock and depart through the shipping dock. The dock locations are indicated by the black circles in Figure 20.26. Assume rectilinear travel distance between the docks and the centroid of the storage bays. All material handling operations are executed with single command material handling cycles. A total of 48 storage bays are available and each storage bay measures 20 by 20 feet. The number of bays required for storage, the number of operations per day, and the arrival day for each product are given in the Table 20.10. The warehouse is operating as a stationary, cyclical process. The arrival day indicates the day in the warehouse cycle that the product gets replenished.

Table 20.10. Product Information

Product	Storage Bays	Demand per Day	Arrival Day
Variable	q	d	r
A	10	5	1
B	8	2	3
C	30	10	2

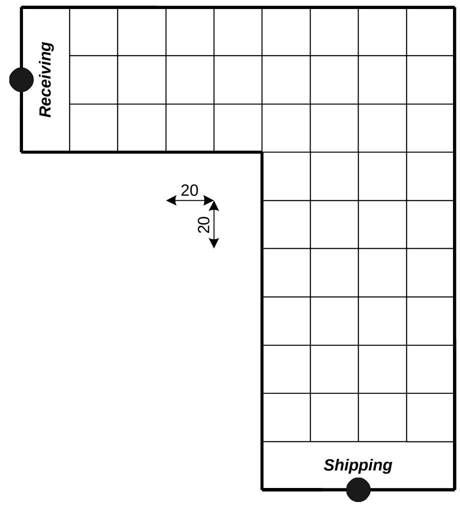


Figure 20.27. Warehouse Layout Distances

Determine the optimal product dedicated storage layout that minimizes the expected travel distance per day. Compute the expected travel time per product and the total expected travel time for this warehouse layout. Assume that the products are stored by decreasing demand in the most desirable locations. Compute the expected travel time per product and the total expected travel time for this warehouse layout. Assume that the products are stored by increasing required storage in the most desirable locations. Compute the expected travel time per product and the total expected travel time for this warehouse layout. What are the required space penalty and travel time penalty (i.e. excess over the best policy) for each policy for this case. Summarize your answer in a clear table.

Assume that the arrival day for products B and C are swapped, i.e. product B now arrives on day 3 and product C now arrives on day 1 of the warehouse cycle. What are the required space penalty and travel time penalty (i.e. excess over the best policy) for each policy for this case. Summarize your answer in a clear table. Discuss any differences or similarities between the previous two tables and explain the reason for the similarities and differences.

Storage Policy Conclusions

Real Systems are Not Perfectly Balanced

Duration of Stay Reduces Travel and Storage Space

2 Class Product Performs Well

Savings Magnitude Depends on Replenishment Pattern

Data Requirements Indicate Automated Warehouses

20.2. Pick versus Reserve Storage Policies

Introduction

Many warehouses are divided into two distinct functional zones. In the first zone, the most frequently demanded items are stored in a storage system where items can be accessed in a high-speed manner. This zone is called the pick zone. Because the storage system is usually expensive, the number of items that can be stored in the pick zone is limited. The second zone holds large quantities of items that are not as frequently accessed. This zone is called the reseve zone. The storage capacity of the reserve zone is for most practical purposes unlimited. Some items can be stored in both the pick and the reserve zone. They are picked from the pick zone and, when necessary, restocked from the reserve zone through an internal replenishment. Typical material handling and storage systems for the pick zone are a flow rack, automated A or V frame order picking systems, and bin shelving. Typical material ahndling and storage systems for the reserve zone are pallet rack and case shelving.

Whenever an item is retrieved from the picking zone rather than from the reserve zone savings are realized. Because the pick zone is small and items stored in it are easily accessible, the cost for picking a single item from it is less than for picking an item from the reserve zone. On other hand, items stored in the pick zone require the extra handling step associated with the internal replenishment. These two costs must be traded off for each product while taking in consideration the total storage capacity of the pick zone. The warehouse manager must then decide if and how much of each product to store in the pick zone.

To make that decision, it is assumed that each product has a dedicated storage space in the pick zone. Products in the pick zone can then be replenished independently of each other, i.e., the pick zone operates under a dedicated storage policy. It is also assumed that the savings in pick costs and the cost of a single internal replenishment of each product is independent of the quantity of the products stored in the pick zone.

The storage capacity of the pick zone and the amount of product stored must be expressed in the same units and represent the critial storage resource of the pick zone. For bin shelving or a gravity flow rack, the critical resource is the area each product takes up of the face of the rack. The storage capacity of the pick zone is then the total rack face area. For an automated A or V frame order picking system, the critical resource is the length along the conveyor belt that a product takes up. The storage capacity of the pick zone is then twice the total length of the pick frame.

Formulation

The following notation will be used:

y_i = binary decision variable indicating if product *i* is stored in the pick zone or not

x_i = continuous decision variable indicating the amount of critical resource space is allocated to item *i*

X = critical storage capacity of the pick zone

N = total number of products in the warehouse

D_i = number of request per unit time for item *i*

R_i = the demand per unit time for item *i* expressed in critical storage units

c_i = cost per internal replenishment of item *i*

e_i = savings per request for item *i* if the item is stored in the pick zone

The decisions of which items and how much of each item to store in the pick zone can then be determined by solving the following formulation.

$$
\begin{aligned}
Max. \quad & \sum_{i=1}^{N} y_i \cdot \left(e_i D_i - \frac{c_i R_i}{x_i} \right) \\
s.t. \quad & \sum_{i=1}^{N} x_i \le X \\
& y_i \in \{0,1\} \\
& x_i \ge 0
\end{aligned}
\tag{20.16}
$$

Observe that the objective function is concave for all positive values of x_i but that the optimal x_i may be zero if it is not profitable to assign item *i* to the pick zone. The above formulation is a typical knapsack problem, which is known to be NP-complete. Hence, it is very unlikely that an efficient optimal solution algorithm can be found for the very large problem instances that typically occur in the pick-versus-reserve problem.

Heuristic

The above problem was studied by Hackman and Rosenblatt (1990). They proposed the following heuristic procedure.

Assume that we know which items are to be stored in the pick zone. In other words, the optimal values of the *y* variables have already been determined. We then need to determine how much of each item to store, subject to the overall capacity constraint. This is the space allocation subproblem.

Let $I^+ = \{ i | y_i = 1 \}$ be the given collection of items that are to be stored in the pick zone, then the optimal quantities to be stored can be determined with the following formulation

$$
\begin{aligned}
Max. \quad & z = \sum_{i \in I^+} \frac{c_i R_i}{x_i} \\
s.t. \quad & \sum_{i \in I^+} x_i \le X \qquad [\, l \,] \\
& x_i > 0
\end{aligned}
\tag{20.17}
$$

The optimal solution to this formulation must satisfy the Kuhn-Tucker conditions, or

$$
l^* = \frac{\partial z}{\partial x_i} = \frac{c_i R_i}{\left(x_i^* \right)^2}
$$

or

$$
x_i^* = \sqrt{\frac{c_i R_i}{l^*}}
$$

Since the objective function is increasing in function of x_i, the capacity constraint will be satisfied as an equality, or

$$
\sum_{i \in I^+} x_i^* = X
$$

which yields

$$
l^* = \left(\frac{1}{X} \sum_{i \in I^+} \sqrt{c_i R_i} \right)^2 \tag{20.18}
$$

and

$$x_i^* = \frac{\sqrt{c_i R_i}}{\sum_{i \in I^+} \sqrt{c_i R_i}} X \tag{20.19}$$

However, the items that are to be located in the pick zone by the optimal solution are not known. Since the original problem is known to be NP-complete and thus difficult to solve to optimality for large problem instances, a heuristic solution will be used. The standard heuristic for solving this problem is to rank the items by the highest "bang-for-the-buck" ratio, i.e., by decreasing ratio of

$$\frac{e_i D_i - \frac{c_i R_i}{x_i}}{x_i} = \frac{e_i D_i}{\sqrt{c_i R_i}} \sqrt{I^*} - I^*$$

The sequence of the items will remain the same if we rank the items based on the non-increasing ratio

$$\frac{e_i D_i}{\sqrt{c_i R_i}} \tag{20.20}$$

If the savings from picking from the active pick area and the cost for the internal replenishment to the active pick area are the same for all products, then this ratio can be further simplied to the following ratio.

$$\frac{D_i}{\sqrt{R_i}} \tag{20.21}$$

In other words, products should be ranked by decreasing ratio of their number of picks divided by the square root of the volume flow over the planning horizon.

Algorithm 20.1. Hackman-Rosenblatt Pick versus Reserve Heuristic

1. Sort items by non-increasing $\frac{e_i D_i}{\sqrt{c_i R_i}}$ ratio. Break ties by placing items with highest denominator first.
2. For each ordered set of items $S_k = \{1, 2, \ldots k\}$, $1 \le k \le N$, compute the optimal space allocation with $x_i^* = \frac{\sqrt{c_i R_i}}{\sum_{i \in I^+} \sqrt{c_i R_i}} X$ and compute the objective function value $z = \sum_{i \in I^+} e_i D_i - \frac{c_i R_i}{x_i}$
3. Keep the set of items S_k with maximum value of *z*. Break ties by selecting the set with the smallest cardinality.

Normally, one would have to solve *N* subproblems to find the best set S_k. However, Hackman and Rosenblatt showed that the function $z(S_k)$ is unimodal with respect to *k*. Hence, a linear search, such as the bisection or Golden Section, reduces the number of subproblems that need to be solved to $O(\log_2 N)$.

Example

In most storage systems the savings per pick and the cost of replenishment are independent of the product being picked or replenished. In addition, the critical resource is usually the volume capacity of the storage system. The following notation will be used:

v_i = fraction of the total storage system volume capacity allocated to product *i*

V = storage system volume capacity

f_i = volume flow of product *i* during the planning horizon, expressed in units such as cubic feet per year

The storage system capacity is then allocated according to the square root of flow

$$v_i^* = \frac{\sqrt{f_i}}{\sum_{i \in I^+} \sqrt{f_i}} V \qquad (20.22)$$

A product with twice the volume flow will get 41 % more space in the storage system. Compare the following two products

Product	A	B
Yearly Demand	5200 units/year	260 units/year
Items per Pick	100	1
Picks per Year	52	260
Volume per Item	4 in^3	64 in^3
Volume Flow per Year	$5200*4/12^3=12.04\ ft^3$	$260*64/12^3=9.63\ ft^3$
Space Fraction	$\frac{\sqrt{12.04}}{\sqrt{12.04}+\sqrt{9.63}} = 0.53$	$\frac{\sqrt{9.63}}{\sqrt{9.63}+\sqrt{12.04}} = 0.47$

Storage Mode Allocation Procedure

This procedure was developed by Bartholdi and Hackman.

Traditionally, the storage mode has been determined based on simple rules that took in consideration the cubic volume and the number of picks of a product. The following figure was presented in Frazelle (1997).

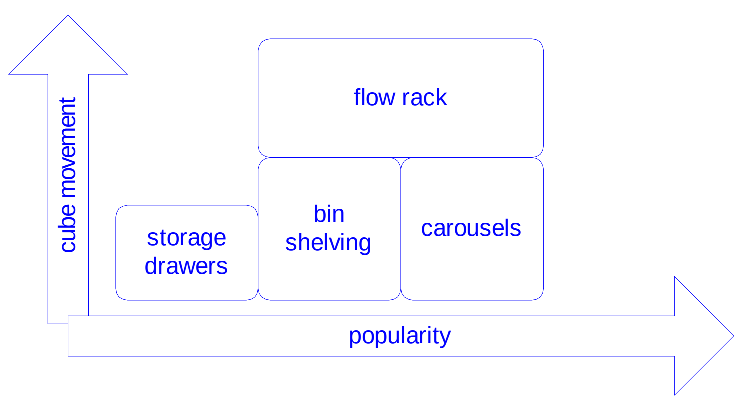


Figure 20.28. Rule-Based Storage Mode Assignment

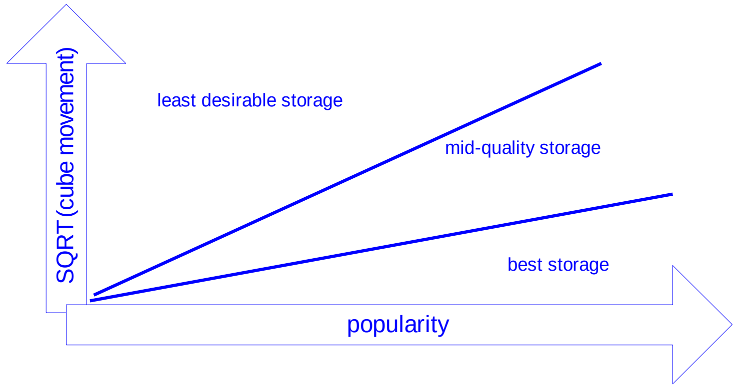


Figure 20.29. Ratio-Based Storage Mode Assignment

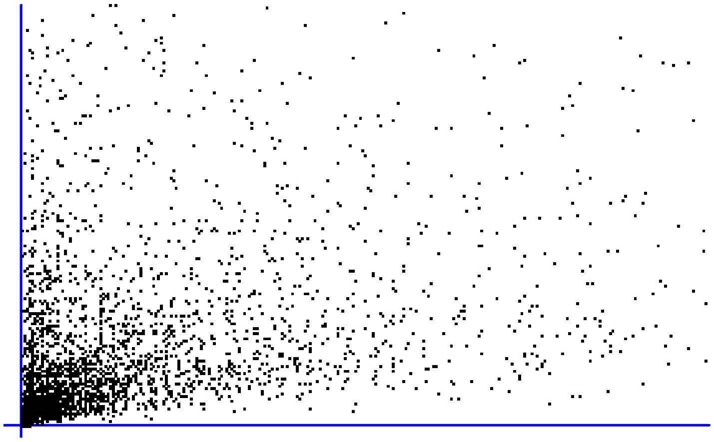


Figure 20.30. Scatter Diagram of 3770 Products

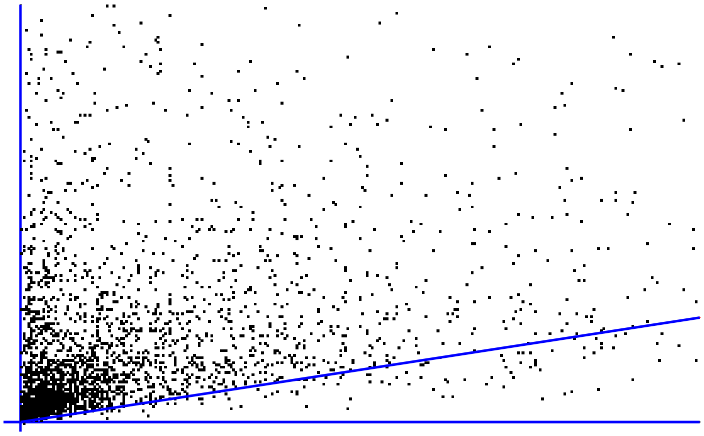


Figure 20.31. Products Allocated to Best Storage Mode

20.3. Block Stacking Storage systems

Introduction

Figure 20.32. Block Stacking Storage Example

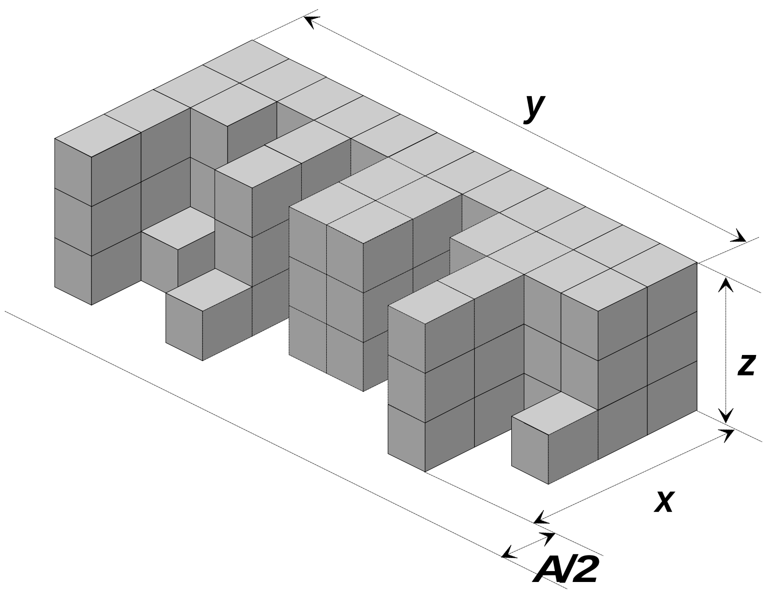


Figure 20.20.33 Block Stacking Illustration

Block Stacking Applications

Few Products

Large Quantities

Palletized or Boxed Products

Without Supporting Rack Structures

Figure 20.34. Block Stacking Storage Example

Figure 20.35. Block Stacking Storage Example

Figure 20.36. Block Stacking Storage Example

Block Stack Characteristics

Stackable Products

High Storage Density

Limited Investment

Limited Product Variety

Block Stacking Terminology

Unit Load
Stack
Lane
Aisle

Single Command Unit Load Storage and Retrieval

Block Stacking Apllication Areas

Finished Goods Warehouse

Distribution Warehouse

Public Warehousing

Block Stacking Objectives

1. Maximize Space Utilization
2. Maximize Storage Flexibility
3. Minimize Transportation Costs

To Determine Optimal or Near-Optimal Lane Depths for Single and Multiple Products That Maximize the Space Utilization in Block Stacking Storage Systems And Minimize Honeycomb Space Loss

Basic Space-Time Tradeoff

Travel Aisle Space + Storage Lane Space versus Time this Space is Occupied

Required Decision Policies

Storage Policy for Arriving Loads in a Product Batch

Warehouse Layout Design

Assumptions

LIFO by Lane

FIFO by Product

No Mixed Lanes

No Relocations

Constant Demand Rate

Instantaneous replenishment

Perfectly Balanced Shared Storage

Perfectly Balanced Shared Storage = Whenever a Lane of Depth is Vacated, a Product Requiring a Lane of that Depth has Arrived

Matson and White (1981,1984) studied extensively the case of a single lane depth for all products in the warehousing system. Goetschalckx and Ratliff (1991) derived a computation procedure for the optimal multiple lane depths in the warehouse and compared this with various heuristic lane depths.

The following notation is used:

Q	=	number of unit loads in batch
W	=	pallet width along the aisle
L	=	pallet length perpendicular to the aisle
A	=	travel aisle width
I	=	safety stock in pallets at time of arrival
d	=	constant demand rate
z	=	stack height in unit loads
x	=	lane depth vector
y	=	number of lanes per depth

N	=	total number of lanes
r_n	=	number of stacks in lanes n+1 through N

The parameters and decision variables are illustrated in the following Figure of a block stacking ground plan.

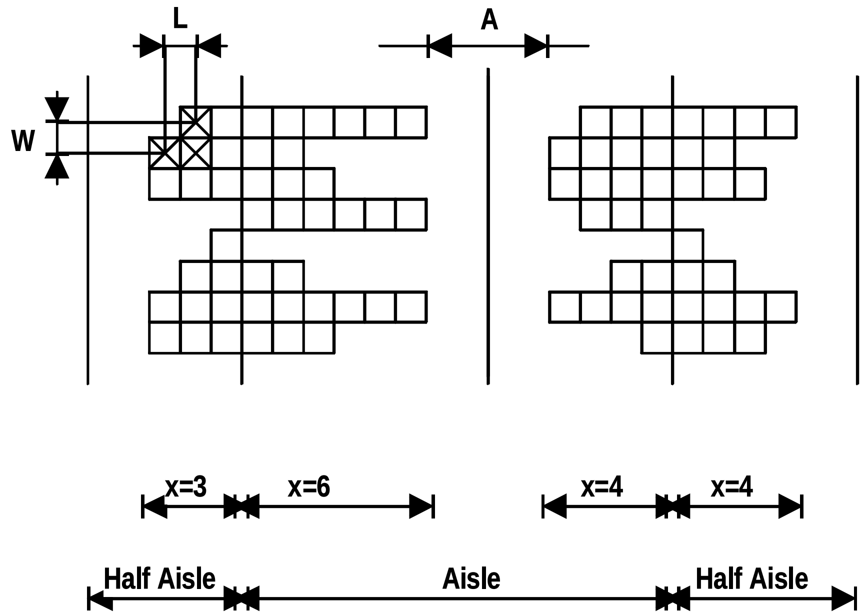


Figure 20.37 Block Stacking Ground Plan

Single Lane Depth Systems

Basic Space-Time Tradeoff

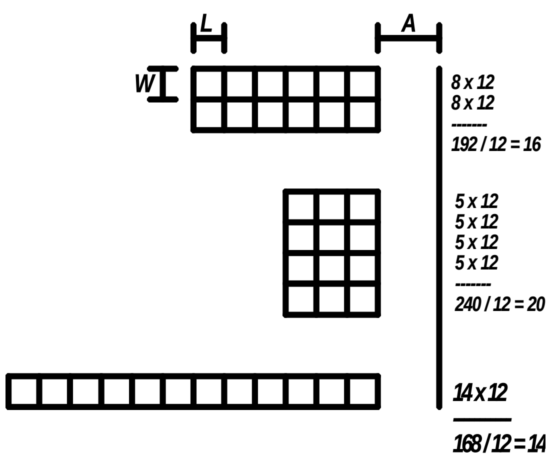


Figure 20.38 Dedicated Storage Time-Space Tradeoff

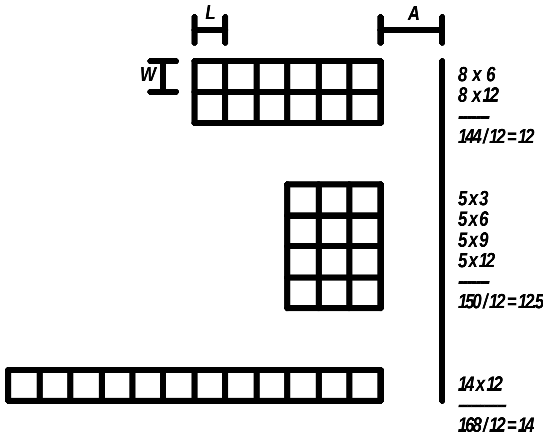


Figure 20.39 Shared Storage Time-Space Tradeoff

Single Lane Depth Derivation for a Single Product

Literature Review

Kind (1965, 1975)

$$x = \sqrt{\frac{QA}{Lz}} - \frac{A}{2L} \tag{20.23}$$

Matson and White (1981, 1984)

$$x = \sqrt{\frac{(Q+2I)A}{2Lz}} \tag{20.24}$$

Optimal Single Lane Depth

Since the number of lanes has to be an integer number, it is computed using the ceiling function, which in essence rounds up the number of lanes if required:

$$y = \left\lceil \frac{Q}{xz} \right\rceil \tag{20.25}$$

The total space-time requirement can be computed by multiplying the square area of each lane with the time this lane is occupied. The footprint area of each lane and its associated aisle space is equal to

$$W(xL + 0.5A) \tag{20.26}$$

The occupation time of the first incomplete lane is:

$$t_1 = \frac{I + [Q - (y-1)zx]}{d}$$

The occupation time for the second through yth lane is then:

$$t_2 = t_1 + \frac{xz}{d}$$

$$t_3 = t_1 + \frac{2xz}{d}$$

$$t_y = t_1 + \frac{(y-1)xz}{d}$$

The sum of the occupation times is then:

$$\sum_{j=1}^{y} t_j = yt_1 + \frac{xz}{d}\sum_{j=1}^{y-1} j$$

$$= \frac{y\left[(Q+I)-(y-1)zx\right]}{d} + \frac{y(y-1)xz}{2d}$$

The total space-time requirement is then given by

$$S = \frac{y(xL+0.5A)W[2(Q+I)-(y-1)xz]}{2d} \qquad (20.27)$$

S is a non-convex function of the lane depth x because x and y both must have integer values. If we consider the continuous relaxation of the problem where x and y no longer have to be integer and the product of xyz is exactly equal to Q, then S_c becomes a convex function of x.

$$S_c = \frac{QW(xL+0.5A)(Q+2I+xz)}{2dxz} \qquad (20.28)$$

Computing the first derivative and setting it equal to zero yields the optimal continuous single lane depth x^*_c. The second derivative is also computed and always larger than zero for non-zero lane depths, which proves that S_c is a convex function of the lane depth x.

$$\frac{dS_c}{dx} = \frac{QW}{d}\left(\frac{L}{2} - \frac{A(Q+2I)}{4x^2z}\right)$$

$$\frac{d^2S_c}{dx^2} = \frac{QWA(Q+2I)}{2dzx^3} > 0$$

$$x_c^* = \sqrt{\frac{(Q+2I)A}{2Lz}} \qquad (20.29)$$

Since S is non-convex for the original problem, the optimal continuous single lane depth is not necessarily the optimal single lane depth. To find the optimal lane depth all possible lane depths are evaluated with complete enumeration. This can be easily done with a spreadsheet. This will be illustrated in the next section for multiple products.

Single Lane Depth Derivation for a Multiple Products

The determination of the optimal single lane depth for multiple products can be best captured in the following table. For each product and for each lane depth x the required number of lanes for the product is computed with Formula 20.25 and the total space-time requirement is computed with Formula 20.27.

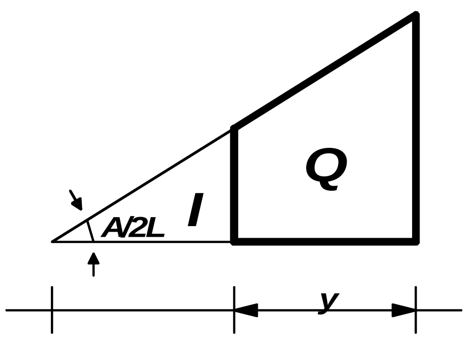
The example considers a warehouse with two products. For both products the length and the width of a pallet including all clearances are equal to 4 feet and the travel aisle is 16 feet wide. For the first product A, the number of pallets in the replenishment batch is equal to 60, the stack height is equal to 3 pallets. The daily demand rate is equal to 0.5 pallets/day. There is no initial safety stock for this product. For the second product B, the number of pallets in the replenishment batch is equal to 60, the stack height is equal to 5 pallets. The daily demand rate is equal to 0.25 pallets/day. There is no initial safety stock for this product. The lane depth computations are shown in the next table. Observe that the best single lane depth for product A is 5 stacks and the best single lane depth for product B is either 4 or 6 stacks, but that the best single lane depth for both products together is 6 stacks.

Table 20.11. Single Lane Depth for Multiple Products

x	y_A	S_A	y_B	S_B	SS_p
1	20	60480	12	74880	135360
2	10	42240	6	53760	96000
3	7	36960	4	48000	84960
4	5	34560	3	46080	80640
5	4	33600	3	47040	80640
6	4	33792	2	46080	79872
7	3	33696	2	48960	82656
8	3	34560	2	51200	85760
9	3	34848	2	52800	87648
10	2	34560	2	53760	88320
11	2	36192	2	54080	90272
12	2	37632	1	53760	91392

Multiple Lane Depths Systems

Goetschalckx and Ratliff (1991) developed a method based on dynamic programming to derive the optimal multiple lane depths for a product.



They compared various methods to derive the lane depths and found that a limited number of lane depths provide a very close performance to the theoretical optimum. The optimal lane depths, selected from a limited number of depths, as computed by the **BLOCK** application, are shown in the next Figure.

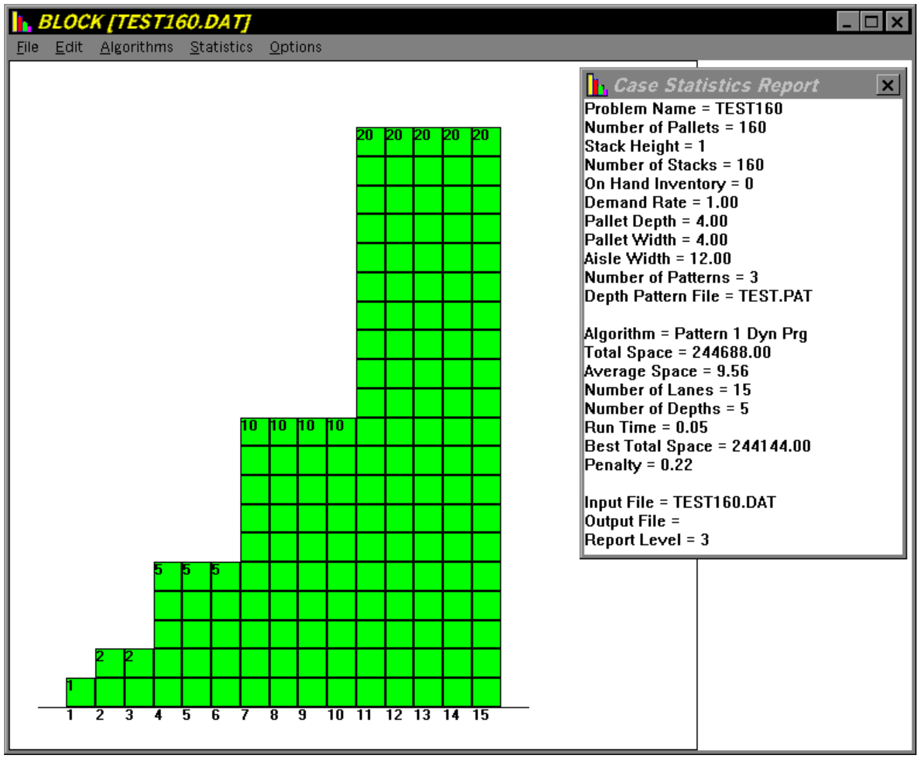


Figure 20.40. BLOCK Application Illustration

Depth Pattern Selection

Maximum 5 or 6 Different Depths

Range with Geometrical Series

Related to Order Batch Sizes

Maximum Depth ≈ $Q / 4z$

Experimental Comparison of Policies

Optimal (GR)

Triangle (TR)

Patterns (P2 & P5)

- P2 = (1, 2, 4, 8, 16, 32)
- P5 = (1, 2, 5, 10, 20, 40)

Discrete Equal (EQ)

Continuous Equal (CE)

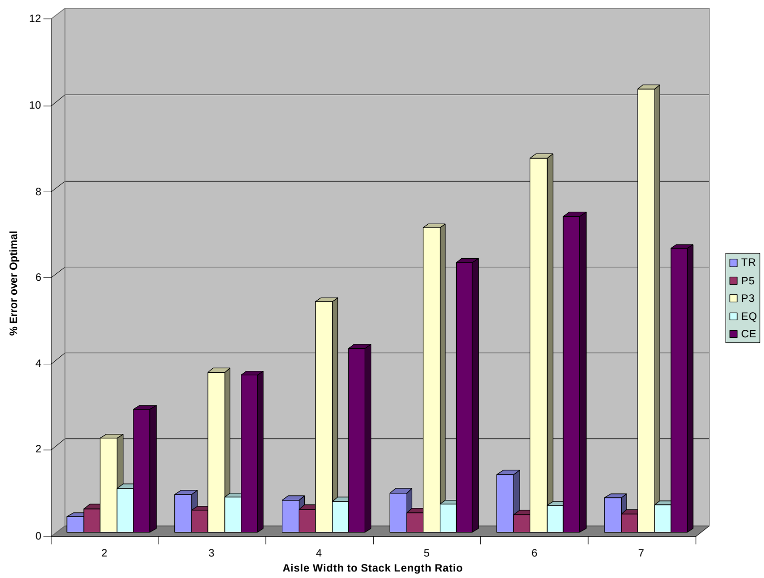


Figure 20.41. Influence of the Aisle to Pallet Ratio on Storage Policy Performance

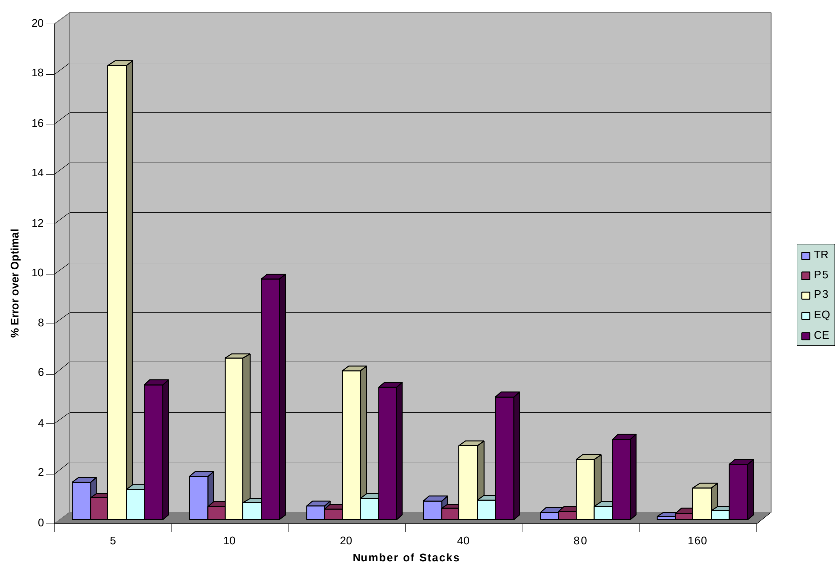


Figure 20.42. Effect of Batch Size on Storage Policy Performance

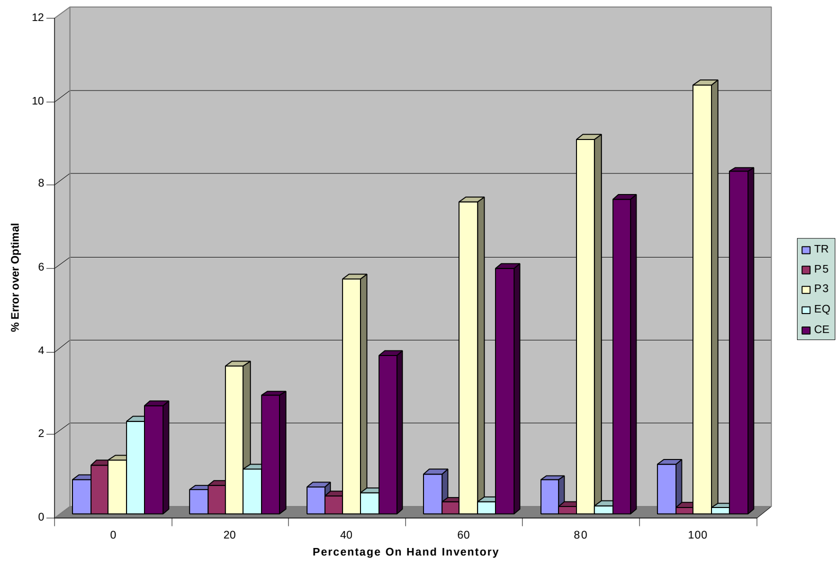


Figure 20.43. Effect on On Hand Inventory on Storage Policy Performance

Warehouse Layout Policy

1. Pick a geometrical series for depth pattern
2. Compute average number of required lanes for each product for each depth
3. Estimate the warehouse sharing factor
4. Compute required number of lanes for each depth
5. Successively round number of lanes to whole aisles for each product
6. Store the arriving batches based on pattern lane depths or in depth-proximity lanes

Deep Lane Storage Systems

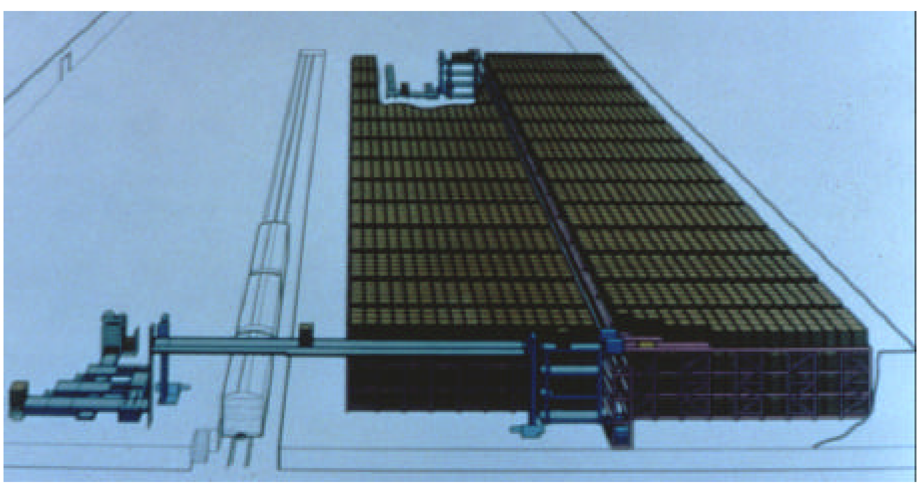


Figure 20.44. Deep Lane Storage System Illustration

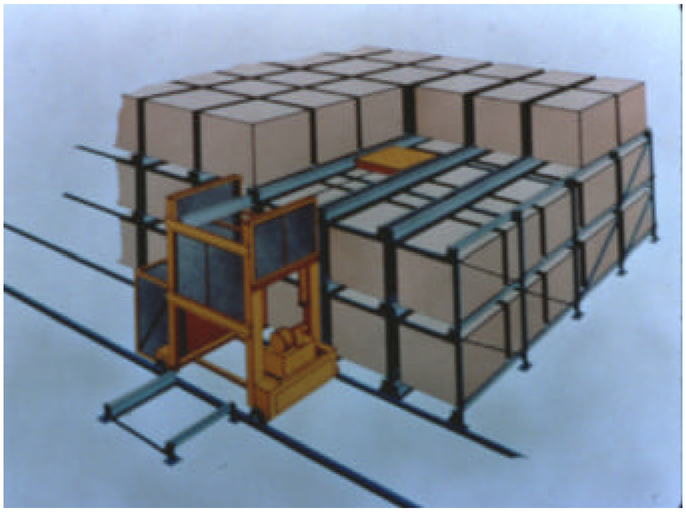


Figure 20.45. Deep Lane Storage Detail Illustration

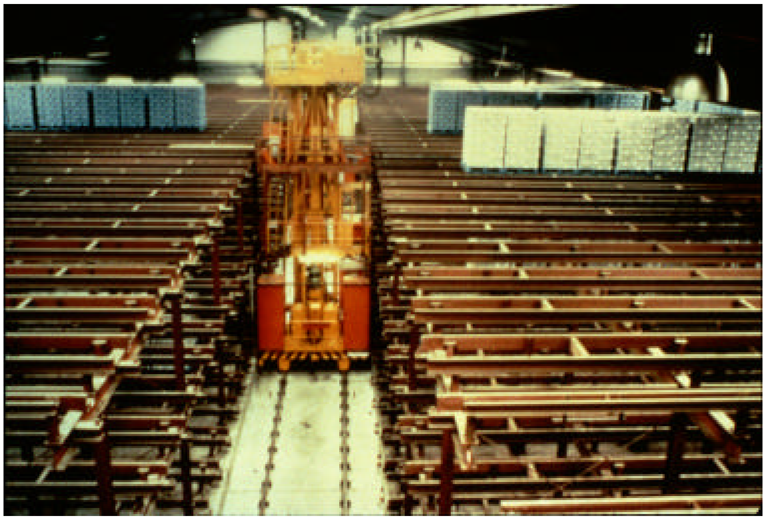


Figure 20.46. Deep Lane Storage System Example