STOCK LOCATION MANAGEMENT

There are numerous storage policies (where the SKUs have to be located) commonly utilized in warehousing. They could be classified as:

Informal **storage policy**

Fixed **(dedicated) storage policy**

Part number **storage policy**

Random **storage policy**

Class based **storage policy**

COI **storage policy**

Shared **storage policy**

…

Also, it has to be pointed that:

each distinct type of load **is termed a** ***Stock Keeping Unit (SKU)*****:**
each different style, size, …
would be assigned a unique SKU.

STOCK LOCATION MANAGEMENT (*contin.*)

Depending on characteristics of warehousing processes and goods, there are used some of policies named:

FIFO **(F**irst **I**n **F**irst **O**ut**)**

LIFO **(L**ast **I**n **F**irst **O**ut**)**

NINO **(N**earest **I**n **N**earest **O**ut**)**

FEFO **(F**irst **E**xpire **F**irst **O**ut**)**

HIFO **(H**ighest [value] **In F**irst **O**ut**)**

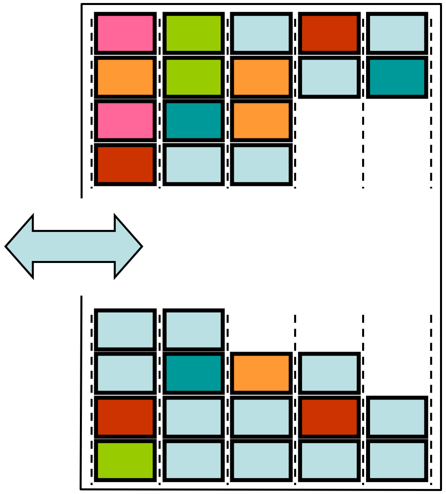
They *are concerned on material handling activities with SKUs* in warehouse and has to be respected in storage policy deciding.

Informal storage policy

This policy is one ***that is not documented***.

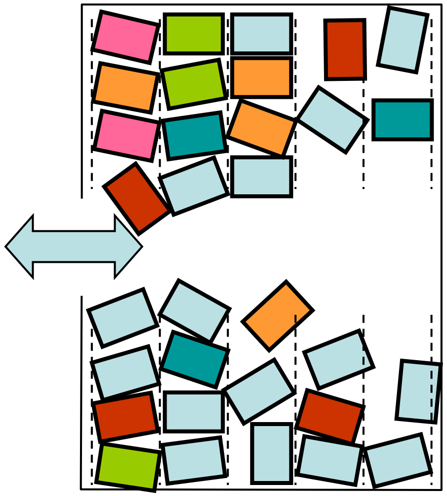
The product is stored wherever is space, and control is very hard. There ***no record is made as to where the product has been stored***.

This *policy relies on the warehouse personnel* having to remember the location and quantity of each SKU.



Advantages:
no maintenance;
high flexibility;
no education necessary

Typically, after an period storage area is looking as …



Disadvantages:
difficult to locate product;
dependent on one person’s knowledge;
inefficient (multiplied tasks) during storing, picking&shipping, damages

Fixed (dedicated) storage policy

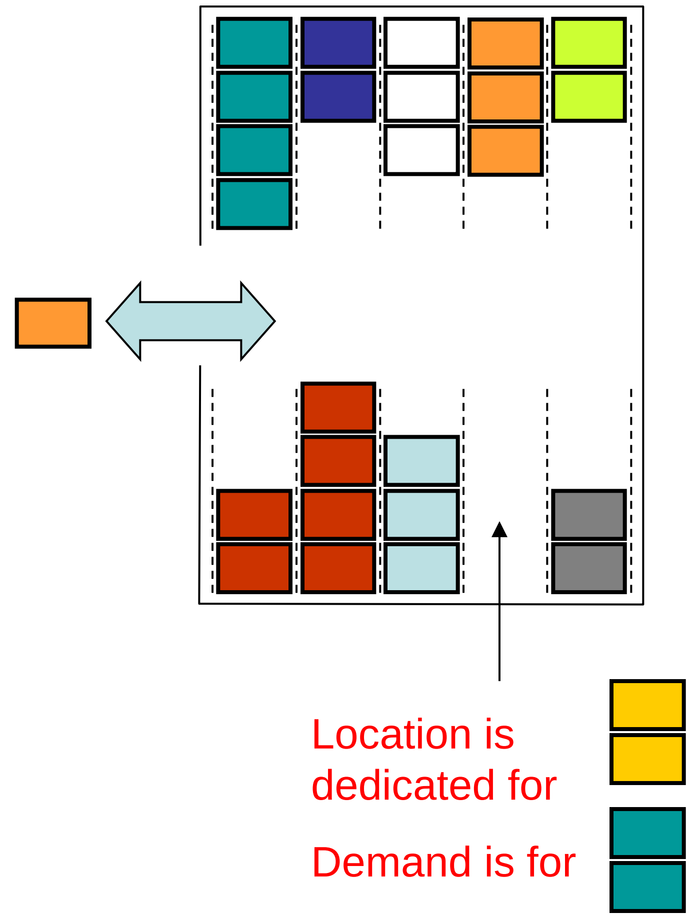
This policy is based on rule that ***every part has its location and every location has its part***. Every SKU is assigned a unique and specific location within warehouse.

It is best suited to a small operation with a small warehouse, small number of SKUs and employees.

If information system provides guidance for picking and putaway, it is adaptable for larger size operations.

Advantages:
easy to establish;
easy to maintain

Disadvantages:
poor space utilization;
must leave space for products not in stock;
must size locations for max. stock;
difficult to expand

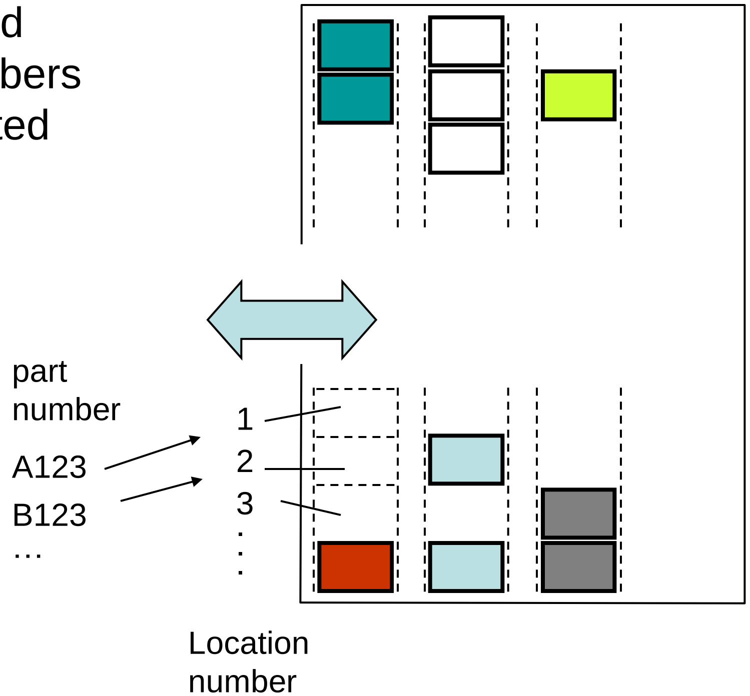


What to do?

Part number storage policy

This policy is actually ***hybrid of dedicated policy***.

The difference is that SKUs are assigned fixed locations based on sequence of the part numbers (A123 comes before B123). This policy is suited for companies with ***consistent part number offering and consistent demand***

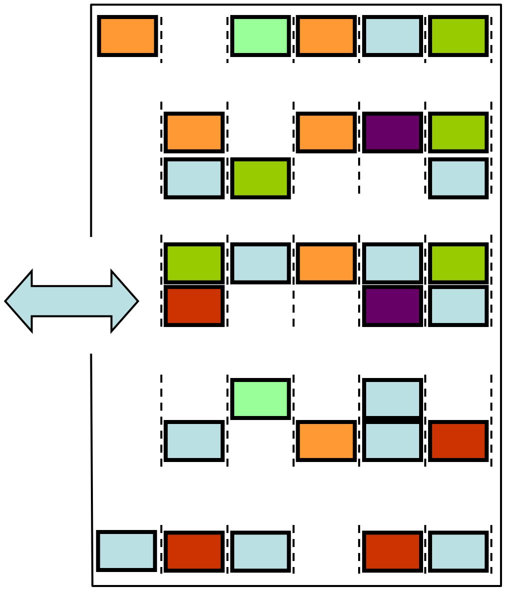


Advantages:
easy to find a product;
product readily accessible;
easy to establish

Disadvantages:
not flexible;
difficult to adjust for demand volumes;
addition of products requires movement of all subsequent items;
poor space utilisation

Random storage policy

The product is ***stored in any, typically closest available location***.
The ***information system has to support*** this policy adequately. The records must be accurately and on time updated with any:
– additional locations,
– location movement/changes of SKU or
– emptying of location through picking.



Advantages:
excellent space utilization;
flexibility; easy to expand;
easy to understand

Disadvantages:
requires detailed records;
discipline to maintain records

Well, discuss a ***random*** and ***dedicated*** storage policy for 4 type of SKUs (A, B, C,D):

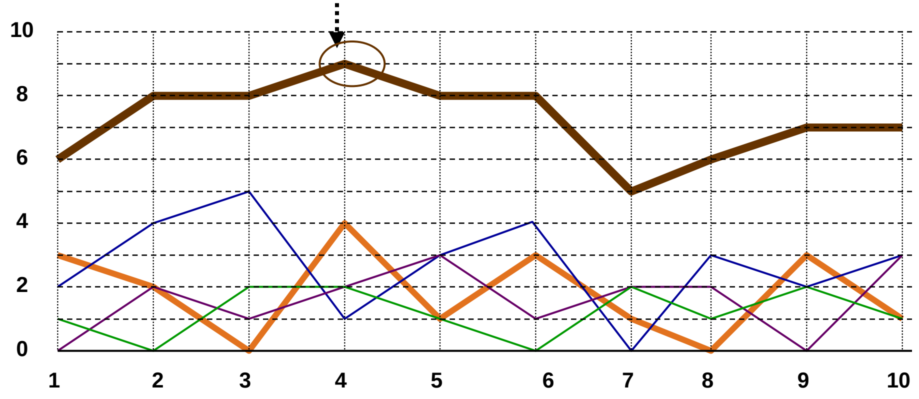
Here is an example of daily inventory level for SKU A, B,C and D and Inventory level in time for sum(A,B,C,D)

(table and graphical presentation)

time	SKU A	SKU B	SKU C	SKU D	SUM(ABCD)
1	3	2	0	1	6
2	2	4	2	0	8
3	0	5	1	2	8
4	4	1	2	2	9
5	1	3	3	1	8
6	3	4	1	0	8
7	1	0	2	2	5
8	0	3	2	1	6
9	3	2	0	2	7
10	1	2	3	1	7
	max(A) = 4	max(B) = 5	max(C) = 3	max(D) = 2	max(sum) = 9

sum(max(i))=14

Necessary storage space for ***random*** policy is
Space (rnd) =
= max(sum(A+B+C+D)) = 9



Necessary storage space for ***dedicated*** policy is
Space (ded) = sum(max(A,B,C,D)) =
4 + 5 + 3 + 2 = 14

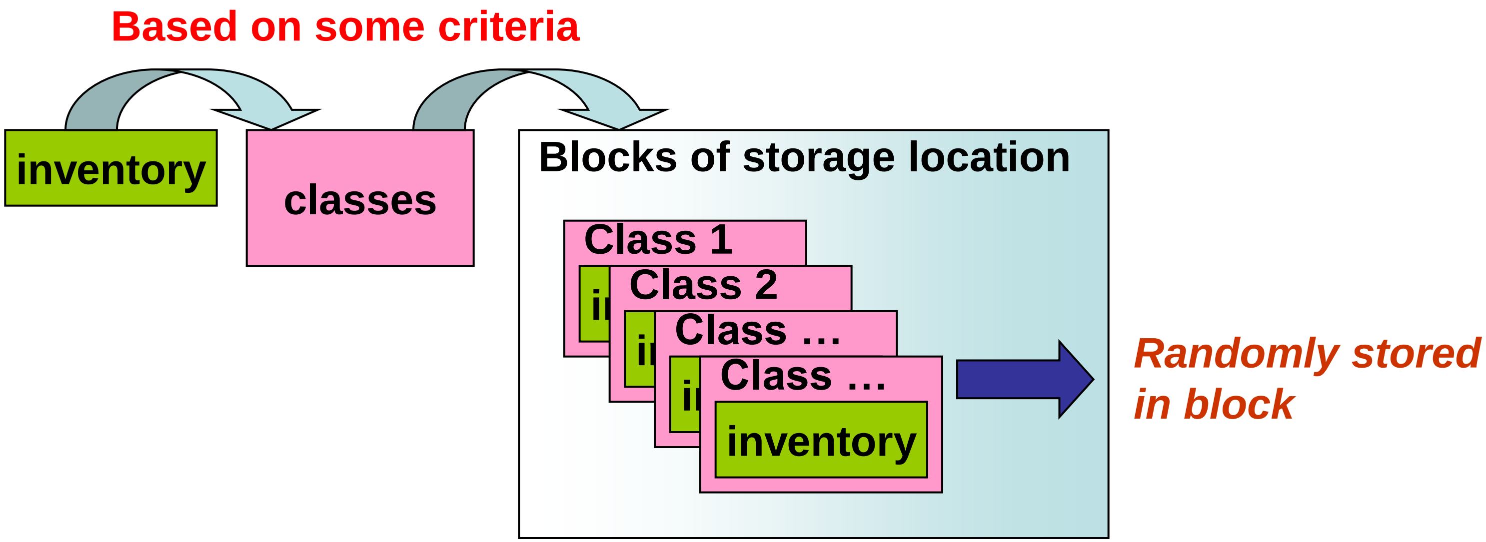
Class based storage policy

It is based on Pareto's effect – in warehouse typically ***80%*** of the S/R activity is directed at ***20% of the items***, ***15%*** at ***30%*** of the items, and the remaining ***5%*** of the S/R activity at ***50%*** of the items (ABC classes).

Class A items must be stored closest to the input/output point(s), class B next closest, and so on.

Although ***each class of items has a dedicated storage space***, ***an item can be stored randomly in any available storage space in the location assigned to its class***, just as any item in a dedicated policy can be randomly stored in any available space dedicated to that item.

The random and dedicated policies are the two extremes, and the COI and class-based policies fall somewhere in between.



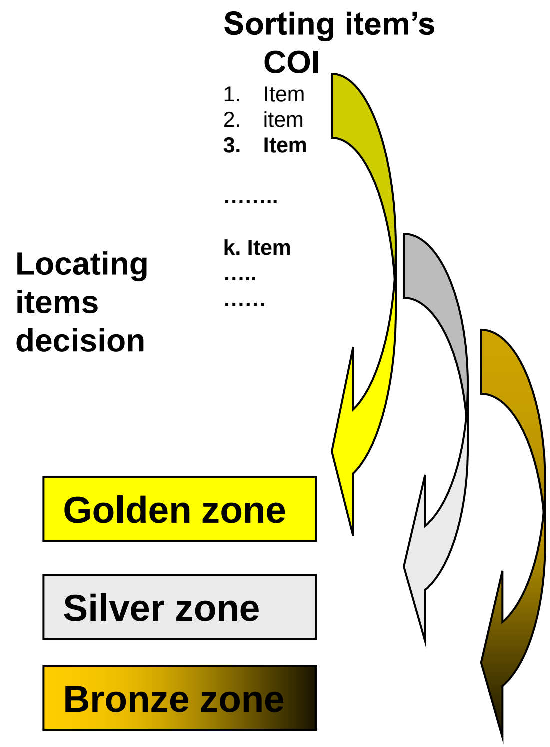
COI (Cube per Order Index) storage policy

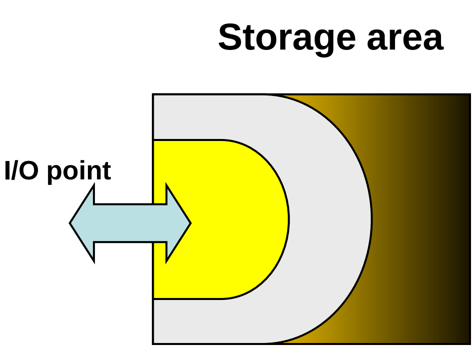
This policy is operationally very simple and is widely used.
The ***COI*** for an item is defined as
the ratio of the item's storage space requirement (its cube) to the number of S/R transactions for that item.

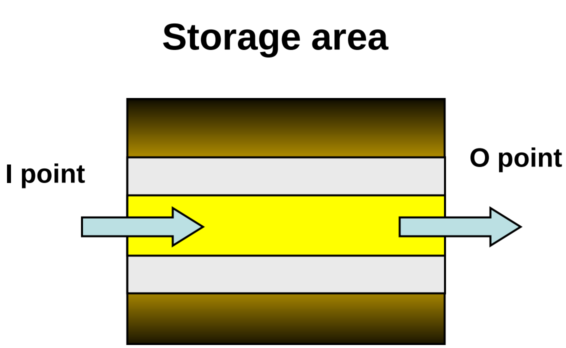
$$COI = \frac{\text{Item's cube (volume)}}{\text{number of transactions}}$$

According to the COI policy a warehouse manager

1. *lists the items in a no decreasing order* of their COIs,
2. *allocates the first item* in the list to the required number of storage spaces that are *nearest* (distance and height) to the input/output (I/O) point,
3. *allocates the second item* in the list to the required number of storage spaces that are *next closest* to the U/O point, *and so on*, until all the items are allocated.
   Thus the COI policy puts items that have a large number of S/R requests and require less storage space near the I/O point.







Shared storage policy

This policy also falls between the two extremes - random storage and dedicated storage. ***As in the random storage operating policy, the same storage space may hold different (subset of) items over time;***
however, the allocation of items to storage spaces is not random, but carefully controlled.

Fast-moving items are stored in spaces closer to the I/O point. Slow-moving items are stored in spaces farther away from the I/O point.
Because items may not be replenished instantaneously but at a constant rate, the time spent in inventory may vary from lot to lot even for the same product.

Also *because different items may reach their maximum inventory levels at different times*, the proper allocation of items to storage locations based on the shared storage policy *can increase system throughput and improve space utilization*.

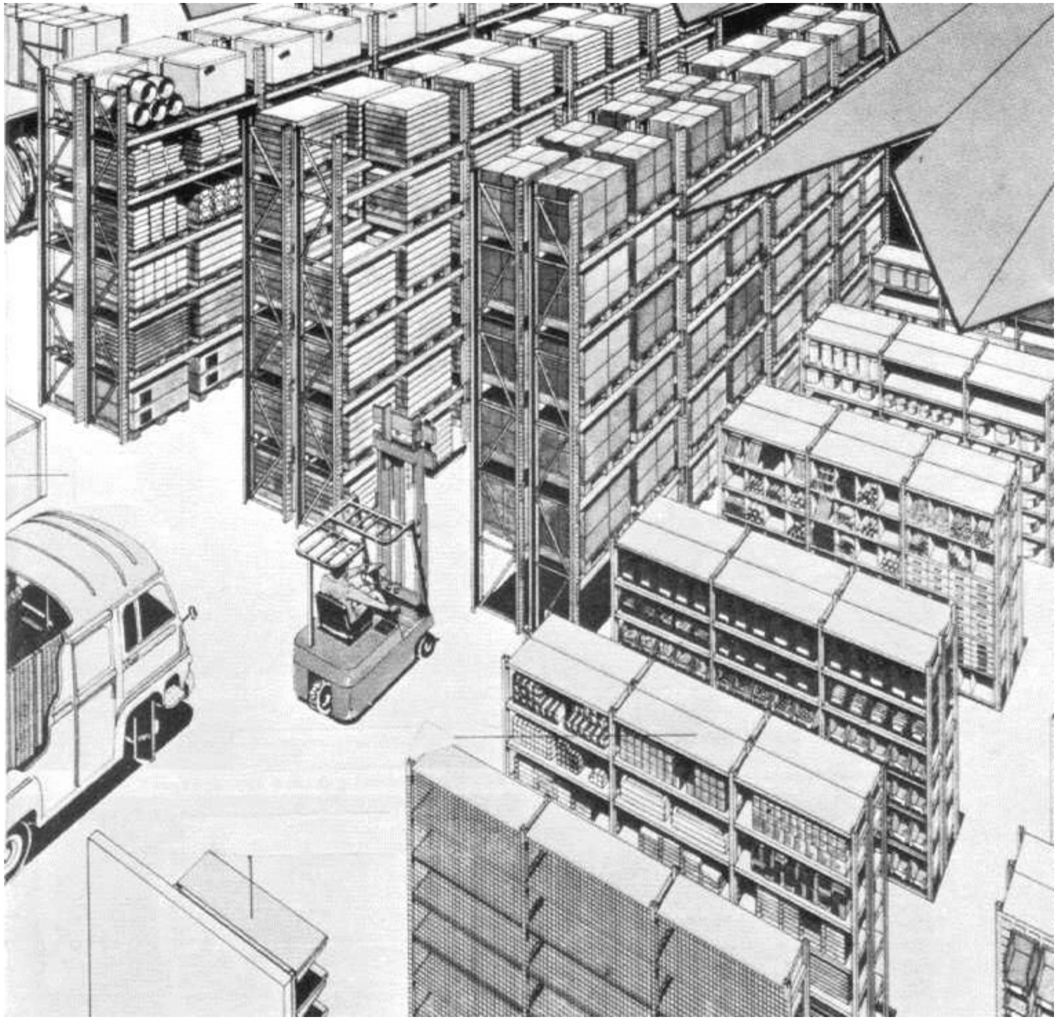
In practice, once a SKUs are grouped together appropriately, decisions to be made on where the product should be located.
There are also a few of common philosophies:

- ***Locate based on popularity***
- ***Locate based on similarity***
- ***Locate based on size***
- ***Consider special characteristics and handling***

or some others

Impact factors on stock location management are also the assortment of products, number of different storage zones, types of SKUs, facility limits, speed of movement, materials handling system, and many others.

Next figures show some solutions of WH storage area



Models for Storage Policies

There are a lot of models present in literature for mentioned storage policies.

Here is briefly explained one - ***Model of Dedicated Storage Policy*** (Heragu)

Consider the following warehouse storage problem:

- a warehouse has p I/O points through which m items enter and leave the warehouse,
- the items are stored in one of n storage spaces or locations,
- each location requires the same storage space, and it is known
- that item i requires S_i storage spaces.

Ideally would be:

$$\sum_{i=1}^{m} S_i = n$$

There are

- f_{ik} trips of item i through I/O point k,
- c_{ik} is the cost of moving a unit load of item i a unit distance through I/O point k, and
- d_{kj} is the distance of storage space j from I/O point k.

Variable x_{ij} specifies whether or not item i is assigned to storage space j. The model is formulated to assign the items to storage spaces in a way that minimizes the cost of moving the items in and out of the I/O points.

Minimize $$\sum_{i=1}^{m} \sum_{j=1}^{n} \left[\frac{\sum_{k=1}^{p} c_{ik} f_{ik} d_{kj}}{S_i} \right] x_{ij} \qquad (1)$$

subject to

$$\sum_{j=1}^{n} x_{ij} = S_i \quad i = 1,2,\ldots,m \qquad (2)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1,2,\ldots,n \qquad (3)$$

$$x_{ij} = 0 \text{ or } 1 \quad i = 1,2,\ldots,m,\ j = 1,2,\ldots,n \qquad (4)$$

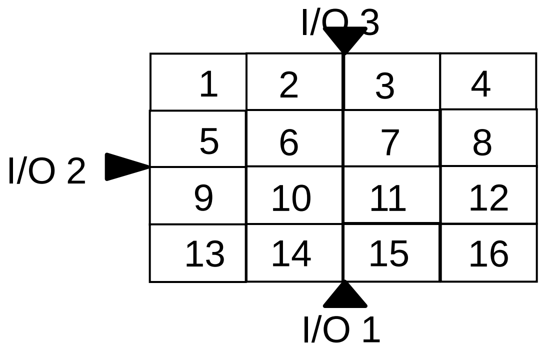
Substituting

$$w_{ij} = \frac{\sum_{k=1}^{p} c_{ik} f_{ik} d_{kj}}{S_i} \qquad (5)$$

objective function from previous slide could be written as:

Minimize $\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_{ij}$

The model is similar to QAP (quadratic assignment problem) except the objective function is linear and the first “assignment” constraint is more general – it is called generalized assignment problem. Because generalized assignment problem is a special case of the transportation model [in constraint (3) instead 1 on the right side is B_j], it can be solved using transportation algorithm. Using that, even we have several thousand storage spaces, the problem can be solved quickly on a computer.



Number of each stock location $(x_1, \ldots, x_{16})$

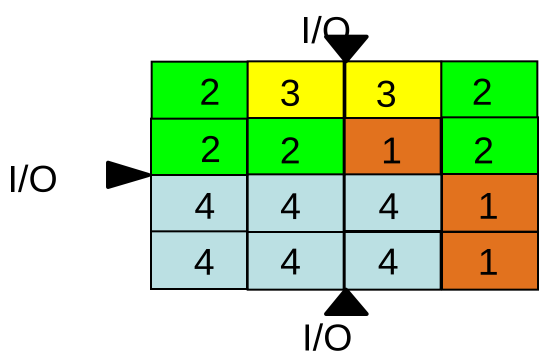
$$[f_{ik}(c_{ik})] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \left[\begin{array}{ccc} 1 & 2 & 3 \\ 150(5) & 25(5) & 88(5) \\ 60(7) & 200(3) & 150(6) \\ 96(4) & 15(7) & 85(9) \\ 175(15) & 135(8) & 90(12) \end{array} \right] \begin{array}{c} S_i \\ 3 \\ 5 \\ 2 \\ 6 \end{array}$$

$[d_{kj}] =$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	5	4	4	5	4	3	3	4	3	2	2	3	2	1	1	2
2	2	3	4	5	1	2	3	4	1	2	3	4	2	3	4	5
3	2	1	1	2	3	2	2	3	4	3	3	4	5	4	4	5

W_{ij}

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1626.7	1271.7	1313.3	1751.7	1481.7	1126.7	1168.3	1606.7	1378.3	1023.3	1065	1503.3	1316.7	961.7	1003.3	1441.7
2	1020	876	996	1380	996	852	972	1356	1092	948	1068	1452	1308	1164	1284	1668
3	1830	1308	1360.5	1987.5	1968	1446	1498.5	2125.5	2158.5	1636.5	1689	2316	2401.5	1879.5	1932	2559
4	2907.5	2470	2650	3447.5	2470	2032.5	2212.5	3010	2212.5	1775	1955	2752.5	2135	1697.5	1877.5	2675



Optimal stock location