cooredinated\_ordering2

Group B

30/05/2020

## Coordinated Ordering

1. Deterministic/continous Demand: The same amount of demand is there every instance of time and demand are met at every instance of time, thus no time varying demand
2. Instantaneous replenishment: As soon as you place an order, the item arrives. Lead time= 0
3. No shortage: As soon as stock level on Y axis reaches 0, place an order for another quantity q. thus instanstaneous replenishment.

In the data set “Data\_ordering.xlsx” parts are represented by material ID column. denotes part $i \in \mathcal I \qquad \{i\_1,i\_2,...,i\_{62}\}, \space$ demand per day for part $i, \space$

# data Extraction

library("readxl")  
product\_data <- read\_excel("Data\_ordering.xlsx",sheet = "product data")  
  
product\_data

## # A tibble: 62 x 5  
## `material ID` `demand per day` `box ID` `pieces/box` price  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 7305667+74 12 6203060 45 1.51  
## 2 7305669+77 30 6203059 15 7.62  
## 3 7305670+77 30 6203059 15 1.62  
## 4 7305673+76 30 6203059 16 1.51  
## 5 7305674+76 30 6203059 16 1.51  
## 6 7305817+74 60 6203060 16 4.55  
## 7 7305819+79 30 6203059 6 11.1   
## 8 7305820+79 30 6203059 6 2.86  
## 9 7305823+73 30 6203059 30 4.05  
## 10 7305824+73 30 6203059 30 2.56  
## # ... with 52 more rows

box\_data <- read\_excel("Data\_ordering.xlsx",sheet = "box data")  
  
box\_data

## # A tibble: 9 x 6  
## `box ID` length width height sorting `ordering cost (€)`  
## <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 3103147 297 198 147 width 50  
## 2 3104147 396 297 147 length 45  
## 3 3106147 594 396 147 length 55  
## 4 3108210 794 596 213 width 65  
## 5 3106410 594 396 280 length 60  
## 6 6203061 297 198 147 width 50  
## 7 6203060 396 297 147 length 75  
## 8 6203059 594 396 147 length 80  
## 9 6203062 594 396 280 length 70

Total\_racks=8  
levels\_per\_rack=4  
rack\_length= 6000  
rack\_width=1750  
rack\_height=300  
  
box\_data$`box ID`

## [1] 3103147 3104147 3106147 3108210 3106410 6203061 6203060 6203059 6203062

# Convert to Boxes

Convert demand , quantity and unit price all from part to box which is the capacity of a box for part

Demand for box with part Price for box with material $de\_i = demand\\_per\\_year \space ,$

#assuming there are 365 working days in a year  
#converting everything to boxes  
  
product\_data$demand\_per\_year= ceiling((product\_data$`demand per day` \*365)/ product\_data$`pieces/box`)  
  
product\_data$box\_cost= product\_data$`pieces/box` \* product\_data$price  
  
product\_data

## # A tibble: 62 x 7  
## `material ID` `demand per day` `box ID` `pieces/box` price demand\_per\_year  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 7305667+74 12 6203060 45 1.51 98  
## 2 7305669+77 30 6203059 15 7.62 730  
## 3 7305670+77 30 6203059 15 1.62 730  
## 4 7305673+76 30 6203059 16 1.51 685  
## 5 7305674+76 30 6203059 16 1.51 685  
## 6 7305817+74 60 6203060 16 4.55 1369  
## 7 7305819+79 30 6203059 6 11.1 1825  
## 8 7305820+79 30 6203059 6 2.86 1825  
## 9 7305823+73 30 6203059 30 4.05 365  
## 10 7305824+73 30 6203059 30 2.56 365  
## # ... with 52 more rows, and 1 more variable: box\_cost <dbl>

Append Ordering cost for each item

#box\_data$`box ID`[1]  
order\_cost <- double(length(product\_data$`box ID`))  
  
 for(j in 1: length(box\_data$`box ID`)){  
 for (k in 1:length(product\_data$`box ID`)) {  
 if(box\_data$`box ID`[j]==product\_data$`box ID`[k]){  
 order\_cost[k] <- box\_data$`ordering cost (€)`[j]  
 }  
 }  
   
 }  
  
product\_data$ordering\_cost <- order\_cost   
product\_data

## # A tibble: 62 x 8  
## `material ID` `demand per day` `box ID` `pieces/box` price demand\_per\_year  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 7305667+74 12 6203060 45 1.51 98  
## 2 7305669+77 30 6203059 15 7.62 730  
## 3 7305670+77 30 6203059 15 1.62 730  
## 4 7305673+76 30 6203059 16 1.51 685  
## 5 7305674+76 30 6203059 16 1.51 685  
## 6 7305817+74 60 6203060 16 4.55 1369  
## 7 7305819+79 30 6203059 6 11.1 1825  
## 8 7305820+79 30 6203059 6 2.86 1825  
## 9 7305823+73 30 6203059 30 4.05 365  
## 10 7305824+73 30 6203059 30 2.56 365  
## # ... with 52 more rows, and 2 more variables: box\_cost <dbl>,  
## # ordering\_cost <dbl>

ordering cost for part $i ,\space$

stock holding cost rate based on unit price and interest rate .

## Separate Ordering (SO)

calculate EOQ for each part i:

$$ q\_i^\* = \sqrt \frac {2 \cdot de\_i \cdot (c\_i^{or}+c^{-or})} {pr\_i \cdot h}$$

#EOQ  
#dei= demand\_per\_year, cori= ordering cost for box i  
#cord= ordering cost, pri= box cost, h= interest rate  
so\_eoq\_fun <- function(dei,cori,cord,pri,h){  
 eoq<- sqrt((2\*dei\*(cori+cord))/(pri\*h))  
   
 return(eoq)  
}  
  
vec\_so\_eoq\_fun <- Vectorize(so\_eoq\_fun(dei=product\_data$demand\_per\_year,cori=product\_data$ordering\_cost,cord=1500,pri=product\_data$box\_cost,h=0.10))

## Warning in formals(fun): argument is not a function

#Append Vector eoq to the table  
product\_data$eoq <- ceiling(vec\_so\_eoq\_fun)

no of orders for part average inventory for part

Note: In row 5 $MateiralID $7305673+76 has 16 pieces in a box and row 8 $MateiralID->$7305819+79 has 6 pieces in a box , both has same $boxID $6203059 with

#dei=product\_data$demand\_per\_year,cori=product\_data$ordering\_cost,cord=1500,pri=product\_data$box\_cost,h=0.10  
  
so\_obj\_funct<-function(cord,cori,dei,eoq,h,pri){  
   
 obj <- (62\*cord)+(sum((dei/eoq)\*cori))+ (h\*sum((eoq/2)\*pri))  
 return(obj)  
}  
  
so\_obj\_funct(1500,product\_data$ordering\_cost,product\_data$demand\_per\_year,product\_data$eoq,0.10,product\_data$box\_cost)

## [1] 186649.1

#library(kableExtra)  
#library(knitr)  
#df <- data.frame(Cat = c("box 1i", "box 2i"),   
 #part\_j1 = linebreak(c("yes", "no")),   
 #part\_j2 = linebreak(c("yes", "no")),  
 #part\_j3=linebreak(c("no","yes")),  
 #part\_jn= linebreak(c("no", "yes")))  
#kable(df, col.names = c("boxID vs MatID", "part\_j1", "part\_j2","part\_j3","part\_jn"), escape = F, caption = "parts each boxes can contain") %>%  
 #kable\_styling(latex\_options = "hold\_position")

capacity of a box for part given in pieces Size of box

Assuming the number of boxes for part , if it takes 4 days for another other to be placed.

# Optimum Ordering constraint

$$q\_i=Number \space boxes \space ordered \\
de\_i=Number \space boxes \space demanded \\
T= \frac {q\_i}{de\_i}=TimeCycle$$

Constraint:

# Rack Length or one runway constraint

Number of boxes in a runway with box ID for part $i\\$: E.g: box ID 3103147 has a width of 198 but Rack length is 6000, maximum number of box 3103147 in runway is $n\_k= \frac {6000}{297}=20.20=20 \space boxes\\$

$$ n\_{k}=\frac {rack\_{length}}{b\_{ji\space length}} \\ $$

lets say that there are 720 boxes with id 3103147 containing part i Mat ID(7346592+72), the first 30 boxes are assigned to runway

number of boxes ordered for part in runway $b\_{ji\space length}$ is the length of box id containing part $i,\space \\$

$$\sum\_{i \in \mathcal I } q\_{ik} \cdot b\_{ji\space length} \le rack\_{length} \qquad \forall k \in \mathcal K $$

# Rack Width

Collapsing Rack 4 levels in a rack and joining the 8 racks to become 1 level.

Total rack width available: Sum of number of parts is equal to box size for part .

$$\sum\_{k \in \mathcal K } q\_{km} \cdot b\_{ji\space width} \le rack\_{total\_{Width}}$$

## Joint Ordering(JO)

Remember each time we order an Item we incur an ordering cost of 1500 € for each of the 62 items.

Joint ordering cost

Let order frequency

Order Frequency:

Therefore Ordering cost is

$holding \_ cost h pr h pr $

# Joint Ordering 2(JO2)

$de\_i=Number \space of \space boxes \space demanded \space for \space item \space i \\$

$q\_i^\*=EOQ \space for \space box \space with \space item \space i \\$

$$ q\_i^\* = \sqrt \frac {2 \cdot de\_i \cdot c\_i^{or}} {pr\_i \cdot h}$$

$$ \frac {de\_i}{q\_i}=Order\space frequency \space for \space item \space i$$

number of unique ordering frequency

e.g there are 62 items out of which 8 unique ordering frequency that is item 1 and item 7 can have the same ordering frequency.